

# The msm Package

November 23, 2006

**Version** 0.7

**Date** 2006-11-21

**Title** Multi-state Markov and hidden Markov models in continuous time

**Author** Christopher Jackson <chris.jackson@imperial.ac.uk>

**Maintainer** Christopher Jackson <chris.jackson@imperial.ac.uk>

**Description** Functions for fitting general continuous-time Markov and hidden Markov multi-state models to longitudinal data. Both Markov transition rates and the hidden Markov output process can be modelled in terms of covariates. A variety of observation schemes are supported, including processes observed at arbitrary times, completely-observed processes, and censored states.

**License** GPL version 2 or newer

## R topics documented:

MatrixExp . . . . .	2
aneur . . . . .	3
boot.msm . . . . .	4
bos . . . . .	6
coef.msm . . . . .	7
crudeinits.msm . . . . .	8
deltamethod . . . . .	9
ematrix.msm . . . . .	11
fev . . . . .	12
hazard.msm . . . . .	13
heart . . . . .	14
hmm-dists . . . . .	14
logLik.msm . . . . .	17
medists . . . . .	18
msm . . . . .	20
odds.msm . . . . .	29
pexp . . . . .	30

plot.msm . . . . .	31
pmatrix.msm . . . . .	32
pmatrix.piecewise.msm . . . . .	34
prevalence.msm . . . . .	35
psor . . . . .	37
qmatrix.msm . . . . .	38
qratio.msm . . . . .	39
sim.msm . . . . .	40
simmulti.msm . . . . .	42
sojourn.msm . . . . .	44
statetable.msm . . . . .	45
msm.summary . . . . .	46
surface.msm . . . . .	47
tnorm . . . . .	48
totlos.msm . . . . .	50
transient.msm . . . . .	51
viterbi.msm . . . . .	52

<b>Index</b>	<b>53</b>
--------------	-----------

---

MatrixExp

*Matrix exponential*

---

## Description

Calculates the exponential of a square matrix.

## Usage

```
MatrixExp(mat, t = 1, n = 20, k = 3, method="pade")
```

## Arguments

mat	A square matrix
t	An optional scaling factor for the eigenvalues of mat
n	Number of terms in the series approximation to the exponential
k	Underflow correction factor, for the series approximation
method	"pade" for the Pade approximation, or "series" for the power series approximation.

## Details

The exponential  $E$  of a square matrix  $M$  is calculated as

$$E = U \exp(D) U^{-1}$$

where  $D$  is a diagonal matrix with the eigenvalues of  $M$  on the diagonal,  $\exp(D)$  is a diagonal matrix with the exponentiated eigenvalues of  $M$  on the diagonal, and  $U$  is a matrix whose columns are the eigenvectors of  $M$ .

This method of calculation is used if  $M$  has distinct eigenvalues. If  $M$  has repeated eigenvalues, then its eigenvector matrix may be non-invertible. In this case, the matrix exponential is calculated using the Pade approximation defined by Moler and van Loan (2003), or the less robust power series approximation,

$$\exp(M) = I + M + M^2/2 + M^3/3! + M^4/4! + \dots$$

For a continuous-time homogeneous Markov process with transition intensity matrix  $Q$ , the probability of occupying state  $s$  at time  $u + t$  conditional on occupying state  $r$  at time  $u$  is given by the  $(r, s)$  entry of the matrix  $\exp(tQ)$ .

The implementation of the Pade approximation was taken from JAGS by Martyn Plummer (<http://www-fis.iarc.fr/~martyn/software/jags>).

The series approximation method was adapted from the corresponding function in Jim Lindsey's R package `rmutil` (<http://popgen.unimaas.nl/~jlindsey/rcode.html>).

## Value

The exponentiated matrix  $\exp(mat)$ .

## References

Cox, D. R. and Miller, H. D. *The theory of stochastic processes*, Chapman and Hall, London (1965)

Moler, C and van Loan, C (2003). Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later. *SIAM Review* **45**, 3–49.

At <http://epubs.siam.org/sam-bin/dbq/article/41801>

---

aneur

*Aortic aneurysm progression data*

---

## Description

This dataset contains longitudinal measurements of grades of aortic aneurysms, measured by ultrasound examination of the diameter of the aorta.

## Usage

`data(aneur)`

**Format**

A data frame containing 4337 rows, with each row corresponding to an ultrasound scan from one of 838 men over 65 years of age.

ptnum	(numeric)	Patient identification number
age	(numeric)	Recipient age at examination (years)
diam	(numeric)	Aortic diameter
state	(numeric)	State of aneurysm.

The states represent successive degrees of aneurysm severity, as indicated by the aortic diameter.

State 1	Aneurysm-free	< 30 cm
State 2	Mild aneurysm	30-44 cm
State 3	Moderate aneurysm	45-54 cm
State 4	Severe aneurysm	> 55 cm

683 of these men were aneurysm-free at age 65 and were re-screened every two years. The remaining men were aneurysmal at entry and had successive screens with frequency depending on the state of the aneurysm. Severe aneurysms are repaired by surgery.

**Source**

The Chichester, U.K. randomised controlled trial of screening for abdominal aortic aneurysms by ultrasonography.

**References**

Jackson, C.H., Sharples, L.D., Thompson, S.G. and Duffy, S.W. and Couto, E. Multi-state Markov models for disease progression with classification error. *The Statistician*, 52(2): 193–209 (2003)

Couto, E. and Duffy, S. W. and Ashton, H. A. and Walker, N. M. and Myles, J. P. and Scott, R. A. P. and Thompson, S. G. (2002) *Probabilities of progression of aortic aneurysms: estimates and implications for screening policy* *Journal of Medical Screening* 9(1):40–42

---

boot.msm

*Bootstrap resampling for multi-state models*

---

**Description**

Draw a number of bootstrap resamples, refit a msm model to the resamples, and calculate statistics on the refitted models.

**Usage**

```
boot.msm(x, stat=pmatrix.msm, B=500, file=NULL)
```

**Arguments**

<code>x</code>	A fitted msm model, as output by <code>msm</code> .
<code>stat</code>	A function to call on each refitted msm model. By default this is <code>pmatrix.msm</code> , returning the transition probability matrix in one time unit. If <code>NULL</code> then no function is computed.
<code>B</code>	Number of bootstrap resamples.
<code>file</code>	Name of a file in which to save partial results after each replicate. This is saved using <code>save</code> and can be restored using <code>load</code> , producing an object called <code>boot.list</code> containing the partial results.

**Details**

The bootstrap datasets are computed by resampling independent transitions between pairs of states (for non-hidden models without censoring), or independent patient series (for hidden models or models with censoring).

Confidence intervals or standard errors for the corresponding statistic can be calculated by summarising the returned list of `B` replicated outputs. This is currently implemented for the transition probability matrix, see `pmatrix.msm`. At the moment, for other outputs, users will have to write their own code to summarise the output of `boot.msm`.

Most of `msm`'s output functions present confidence intervals based on asymptotic standard errors calculated from the Hessian. These are expected to be underestimates of the true standard errors (Cramer-Rao lower bound). Bootstrapping may give a more accurate estimate of the uncertainty.

All objects used in the original call to `msm` which produced `x`, such as the `qmatrix`, should be in the working environment, or else `boot.msm` will produce an "object not found" error. This enables `boot.msm` to refit the original model to the replicate datasets.

If `stat` is `NULL`, then `B` different `msm` model objects will be stored in memory. This is inadvisable, as `msm` objects tend to be large, as they contain the original data used for the `msm` fit, so this will be wasteful of memory.

To specify more than one statistic, write a function consisting of a list of different function calls, for example,

```
stat = function(x) list (pmatrix.msm(x, t=1), pmatrix.msm(x, t=2))
```

**Value**

A list with `B` components, containing the result of calling function `stat` on each of the refitted models. If `stat` is `NULL`, then each component just contains the refitted model. If one of the `B` model fits was unsuccessful and resulted in an error, then the corresponding list component will contain the error message.

**Author(s)**

C.H.Jackson <chris.jackson@imperial.ac.uk>

**References**

Efron, B. and Tibshirani, R.J. (1993) *An Introduction to the Bootstrap*, Chapman and Hall.

**See Also**

[pmatrx.msm](#), [totlos.msm](#)

**Examples**

```
## Not run:
## Psoriatic arthritis example
data(psor)
psor.q <- rbind(c(0,0.1,0,0),c(0,0,0.1,0),c(0,0,0,0.1),c(0,0,0,0))
psor.msm <- msm(state ~ months, subject=ptnum, data=psor, qmatrix = psor.q, covariates = ~
## Bootstrap the baseline transition intensity matrix. This will take a long time.
q.list <- boot.msm(psor.msm, function(x)x$Qmatrices$baseline)
## Manipulate the resulting list of matrices to calculate bootstrap standard errors.
apply(array(unlist(q.list), dim=c(4,4,5)), c(1,2), sd)
## Similarly calculate a bootstrap 95% confidence interval
apply(array(unlist(q.list), dim=c(4,4,5)), c(1,2), function(x)quantile(x, c(0.025, 0.975)))
## Bootstrap standard errors are larger than the asymptotic standard errors calculated from
psor.msm$QmatricesSE$baseline
## End(Not run)
```

---

bos

*Bronchiolitis obliterans syndrome after lung transplants*

---

**Description**

A dataset containing histories of bronchiolitis obliterans syndrome (BOS) from lung transplant recipients. BOS is a chronic decline in lung function, often observed after lung transplantation. The condition is classified into four stages of severity: none, mild, moderate and severe.

**Usage**

```
data(bos)
```

**Format**

A data frame containing 638 rows, grouped by patient, including histories of 204 patients. The first observation for each patient is defined to be stage 1, no BOS, at six months after transplant. Subsequent observations denote the entry times into stages 2, 3, 4, representing mild, moderate and severe BOS respectively, and stage 5, representing death.

ptnum	(numeric)	Patient identification number
time	(numeric)	Months after transplant
state	(numeric)	BOS state entered at this time

**Details**

The entry time of each patient into each stage of BOS was estimated by clinicians, based on their history of lung function measurements and acute rejection and infection episodes. BOS is only

assumed to occur beyond six months after transplant. In the first six months the function of each patient's new lung stabilises. Subsequently BOS is diagnosed by comparing the lung function against the "baseline" value.

### Source

Papworth Hospital, U.K.

### References

Heng, D. et al. (1998). Bronchiolitis Obliterans Syndrome: Incidence, Natural History, Prognosis, and Risk Factors. *Journal of Heart and Lung Transplantation* 17(12)1255–1263.

---

<code>coef.msm</code>	<i>Extract model coefficients</i>
-----------------------	-----------------------------------

---

### Description

Extract the estimated log transition intensities and the corresponding linear effects of each covariate.

### Usage

```
## S3 method for class 'msm':
coef(object, ...)
```

### Arguments

<code>object</code>	A fitted multi-state model object, as returned by <a href="#">msm</a> .
<code>...</code>	(unused) further arguments passed to or from other methods.

### Value

If there is no misclassification, `coef.msm` returns a list of matrices. The first component, labelled `logbaseline`, is a matrix containing the estimated transition intensities on the log scale with any covariates fixed at their means in the data. Each remaining component is a matrix giving the linear effects of the labelled covariate on the matrix of log intensities.

For misclassification models, `coef.msm` returns a list of lists. The first component, `Qmatrices`, is a list of matrices as described in the previous paragraph. The additional component `Ematrices` is a list of similar format containing the logit-misclassification probabilities and any estimated covariate effects.

### Author(s)

C. H. Jackson <chris.jackson@imperial.ac.uk>

### See Also

[msm](#)

---

crudeinits.msm      *Calculate crude initial values for transition intensities*

---

### Description

Calculates crude initial values for transition intensities by assuming that the data represent the exact transition times of the Markov process.

### Usage

```
crudeinits.msm(formula, subject, qmatrix, data=NULL, censor=NULL, censor.states=NU)
```

### Arguments

formula	A formula giving the vectors containing the observed states and the corresponding observation times. For example, state ~ time Observed states should be in the set 1, ..., n, where n is the number of states.
subject	Vector of subject identification numbers for the data specified by formula. If missing, then all observations are assumed to be on the same subject. These must be sorted so that all observations on the same subject are adjacent.
qmatrix	Matrix of indicators for the allowed transitions. An initial value will be estimated for each value of qmatrix that is greater than zero. Transitions are taken as disallowed for each entry of qmatrix that is 0.
data	An optional data frame in which the variables represented by subject and state can be found.
censor	A state, or vector of states, which indicates censoring. See <a href="#">msm</a> .
censor.states	Specifies the underlying states which censored observations can represent. See <a href="#">msm</a> .

### Details

Suppose we want a crude estimate of the transition intensity  $q_{rs}$  from state  $r$  to state  $s$ . If we observe  $n_{rs}$  transitions from state  $r$  to state  $s$ , and a total of  $n_r$  transitions from state  $r$ , then  $q_{rs}/q_{rr}$  can be estimated by  $n_{rs}/n_r$ . Then, given a total of  $T_r$  years spent in state  $r$ , the mean sojourn time  $1/q_{rr}$  can be estimated as  $T_r/n_r$ . Thus,  $n_{rs}/T_r$  is a crude estimate of  $q_{rs}$ .

If the data do represent the exact transition times of the Markov process, then these are the exact maximum likelihood estimates.

Observed transitions which are incompatible with the given qmatrix are ignored. Censored states are ignored.

### Value

The estimated transition intensity matrix. This can be used as the qmatrix argument to [msm](#).



**Author(s)**

C. H. Jackson <chris.jackson@imperial.ac.uk>

**See Also**

[statetable.msm](#)

**Examples**

```
data(heart)
twoway4.q <- rbind(c(-0.5, 0.25, 0, 0.25), c(0.166, -0.498, 0.166, 0.166),
c(0, 0.25, -0.5, 0.25), c(0, 0, 0, 0))
statetable.msm(state, PTNUM, data=heart)
crudeinits.msm(state ~ years, PTNUM, data=heart, qmatrix=twoway4.q)
```

---

deltamethod

*The delta method*


---

**Description**

Delta method for approximating the standard error of a transformation  $g(X)$  of a random variable  $X = (x_1, x_2, \dots)$ , given estimates of the mean and covariance matrix of  $X$ .

**Usage**

```
deltamethod(g, mean, cov, ses=TRUE)
```

**Arguments**

g	A formula representing the transformation. The variables must be labelled $x_1, x_2, \dots$ . For example, $\sim 1 / (x_1 + x_2)$ If the transformation returns a vector, then a list of formulae representing $(g_1, g_2, \dots)$ can be provided, for example <code>list( ~ x1 + x2, ~ x1 / (x1 + x2) )</code>
mean	The estimated mean of $X$
cov	The estimated covariance matrix of $X$
ses	If TRUE, then the standard errors of $g_1(X), g_2(X), \dots$ are returned. Otherwise the covariance matrix of $g(X)$ is returned.

## Details

The delta method expands a differentiable function of a random variable about its mean, usually with a first-order Taylor approximation, and then takes the variance. For example, an approximation to the covariance matrix of  $g(X)$  is given by

$$\text{Cov}(g(X)) = g'(\mu)\text{Cov}(X)[g'(\mu)]^T$$

where  $\mu$  is an estimate of the mean of  $X$ .

A limitation of this function is that variables created by the user are not visible within the formula `g`. To work around this, it is necessary to build the formula as a string, using functions such as `sprintf`, then to convert the string to a formula using `as.formula`. See the example below.

## Value

A vector containing the standard errors of  $g_1(X), g_2(X), \dots$  or a matrix containing the covariance of  $g(X)$ .

## Author(s)

C. H. Jackson <chris.jackson@imperial.ac.uk>

## References

Oehlert, G. W. *A note on the delta method*. *American Statistician* 46(1), 1992

## Examples

```
## Simple linear regression, E(y) = alpha + beta x
x <- 1:100
y <- rnorm(100, 4*x, 5)
toy.lm <- lm(y ~ x)
estmean <- coef(toy.lm)
estvar <- summary(toy.lm)$cov.unscaled * summary(toy.lm)$sigma^2

## Estimate of (1 / (alphahat + betahat))
1 / (estmean[1] + estmean[2])
## Approximate standard error
deltamethod (~ 1 / (x1 + x2), estmean, estvar)

## We have a variable z we would like to use within the formula.
z <- 1
## deltamethod (~ z / (x1 + x2), estmean, estvar) will not work.
## Instead, build up the formula as a string, and convert to a formula.
form <- sprintf("~ %f / (x1 + x2)", z)
form
deltamethod(as.formula(form), estmean, estvar)
```

ematrix.msm

*Misclassification probability matrix***Description**

Extract the estimated misclassification probability matrix, and corresponding confidence intervals, from a fitted multi-state model at a given set of covariate values.

**Usage**

```
ematrix.msm(x, covariates="mean", cl=0.95)
```

**Arguments**

<code>x</code>	A fitted multi-state model, as returned by <a href="#">msm</a>
<code>covariates</code>	The covariate values for which to estimate the misclassification probability matrix. This can either be: <ul style="list-style-type: none"> <li>the string "mean", denoting the means of the covariates in the data (this is the default),</li> <li>the number 0, indicating that all the covariates should be set to zero,</li> <li>or a list of values, with optional names. For example <pre>list (60, 1)</pre> where the order of the list follows the order of the covariates originally given in the model formula, or a named list, <pre>list (age = 60, sex = 1)</pre> </li> </ul>
<code>cl</code>	Width of the symmetric confidence interval to present. Defaults to 0.95.

**Details**

Misclassification probabilities and covariate effects are estimated on the logit scale by [msm](#). A covariance matrix is estimated from the Hessian of the maximised log-likelihood. From these, the delta method is used to obtain standard errors of the probabilities on the natural scale at arbitrary covariate values. Confidence intervals are estimated by assuming normality on the logit scale.

**Value**

A list with components:

<code>estimate</code>	Estimated misclassification probability matrix.
<code>SE</code>	Corresponding approximate standard errors.
<code>L</code>	Lower confidence limits.
<code>U</code>	Upper confidence limits.

The default print method for objects returned by `ematrix.msm` presents estimates and confidence limits. To present estimates and standard errors, do something like

```
ematrix.msm(x) [c("estimates", "SE")]
```

### Author(s)

C. H. Jackson <chris.jackson@imperial.ac.uk>

### See Also

`qmatrix.msm`

---

fev

*FEV1 measurements from lung transplant recipients*

---

### Description

A series of measurements of the forced expiratory volume in one second (FEV1) from lung transplant recipients, from six months onwards after their transplant.

### Usage

```
data(fev)
```

### Format

A data frame containing 5896 rows. There are 204 patients, the rows are grouped by patient number and ordered by days after transplant. Each row represents an examination and containing an additional covariate.

<code>ptnum</code>	(numeric)	Patient identification number.
<code>days</code>	(numeric)	Examination time (days after transplant).
<code>fev</code>	(numeric)	Percentage of baseline FEV1. A code of 999 indicates the patient's date of death.
<code>acute</code>	(numeric)	0/1 indicator for whether the patient suffered an acute infection or rejection within 14 days of the visit.

### Details

A baseline "normal" FEV1 for each individual is calculated using measurements from the first six months after transplant. After six months, as presented in this dataset, FEV1 is expressed as a percentage of the baseline value.

FEV1 is monitored to diagnose bronchiolitis obliterans syndrome (BOS), a long-term lung function decline, thought to be a form of chronic rejection. Acute rejections and infections also affect the lung function in the short term.

**Source**

Papworth Hospital, U.K.

**References**

Jackson, C.H. and Sharples, L.D. Hidden Markov models for the onset and progression of bronchiolitis obliterans syndrome in lung transplant recipients *Statistics in Medicine*, 21(1): 113–128 (2002).

---

hazard.msm	<i>Calculate tables of hazard ratios for covariates on transition intensities</i>
------------	---

---

**Description**

Hazard ratios are computed by exponentiating the estimated covariate effects on the log-transition intensities. This function is called by [summary.msm](#).

**Usage**

```
hazard.msm(x, hazard.scale = 1, cl = 0.95)
```

**Arguments**

x	Output from <a href="#">msm</a> representing a fitted multi-state model.
hazard.scale	Vector with same elements as number of covariates on transition rates. Corresponds to the increase in each covariate used to calculate its hazard ratio. Defaults to all 1.
cl	Width of the symmetric confidence interval to present. Defaults to 0.95.

**Value**

A list of tables containing hazard ratio estimates, one table for each covariate. Each table has three columns, containing the hazard ratio, and an approximate upper and lower confidence limit respectively (assuming normality on the log scale), for each Markov chain transition intensity.

**Author(s)**

C. H. Jackson <chris.jackson@imperial.ac.uk>

**See Also**

[msm](#), [summary.msm](#), [odds.msm](#)

---

heart *Heart transplant monitoring data*

---

### Description

A series of approximately yearly angiographic examinations of heart transplant recipients. The state at each time is a grade of cardiac allograft vasculopathy (CAV), a deterioration of the arterial walls.

### Usage

```
data(heart)
```

### Format

A data frame containing 2846 rows. There are 622 patients, the rows are grouped by patient number and ordered by years after transplant, with each row representing an examination and containing additional covariates.

PTNUM	(numeric)	Patient identification number
age	(numeric)	Recipient age at examination (years)
years	(numeric)	Examination time (years after transplant)
dage	(numeric)	Age of heart donor (years)
sex	(character)	sex (0=male, 1=female)
pdiag	(character)	Primary diagnosis (reason for transplant) IHD=ischaemic heart disease, IDC=idiopathic dilated cardiomyopathy.
cumrej	(numeric)	Cumulative number of acute rejection episodes
state	(numeric)	State at the examination. State 1 represents no CAV, state 2 is mild/moderate CAV and state 3 is severe CAV. State 4 indicates death.

### Source

Papworth Hospital, U.K.

### References

Sharples, L.D. and Jackson, C.H. and Parameshwar, J. and Wallwork, J. and Large, S.R. (2003). Diagnostic accuracy of coronary angiopathy and risk factors for post-heart-transplant cardiac allograft vasculopathy. *Transplantation* 76(4):679-82

---

hmm-dists *Hidden Markov model constructors*

---

## Description

These functions are used to specify the distribution of the response conditionally on the underlying state in a hidden Markov model. A list of these function calls, with one component for each state, should be used for the `hmodel` argument to `msm`. The initial values for the parameters of the distribution should be given as arguments.

## Usage

```
hmmCat(prob, basecat)
hmmIdent(x)
hmmUnif(lower, upper)
hmmNorm(mean, sd)
hmmLNorm(meanlog, sdlog)
hmmExp(rate)
hmmGamma(shape, rate)
hmmWeibull(shape, scale)
hmmPois(rate)
hmmBinom(size, prob)
hmmTNorm(mean, sd, lower, upper)
hmmMETNorm(mean, sd, lower, upper, sderr, meanerr=0)
hmmMEUnif(lower, upper, sderr, meanerr=0)
hmmNBinom(dis, prob)
```

## Arguments

`hmmCat` represents a categorical response distribution on the set  $1, 2, \dots, \text{length}(\text{prob})$ . The Markov model with misclassification is an example of this type of model. The categories in this case are (some subset of) the observed states.

The `hmmIdent` distribution is used for underlying states which are observed exactly without error.

`hmmUnif`, `hmmNorm`, `hmmLNorm`, `hmmExp`, `hmmGamma`, `hmmWeibull`, `hmmPois`, `hmmBinom`, `hmmTNorm` and `hmmNBinom` represent Uniform, Normal, log-Normal, exponential, Gamma, Weibull, Poisson, Binomial, truncated Normal and negative binomial distributions, respectively, with parameterisations the same as the default parameterisations in the corresponding base R distribution functions.

The `hmmMETNorm` and `hmmMEUnif` distributions are truncated Normal and Uniform distributions, but with additional Normal measurement error on the response. These are generalisations of the distributions proposed by Satten and Longini (1994) for modelling the progression of CD4 cell counts in monitoring HIV disease. See [medists](#) for density, distribution, quantile and random generation functions for these distributions. See also [tnorm](#) for density, distribution, quantile and random generation functions for the truncated Normal distribution.

<code>prob</code>	( <code>hmmCat</code> ) Vector of probabilities of observing category $1, 2, \dots, \text{length}(\text{prob})$ respectively. Or the probability governing a binomial or negative binomial distribution.
<code>basecat</code>	( <code>hmmCat</code> ) Category which is considered to be the "baseline", so that during estimation, the probabilities are parameterised as probabilities relative to this baseline category. By default, the category with the greatest probability is used as the baseline.

<code>x</code>	( <code>hmmIdent</code> ) Code in the data which denotes the exactly-observed state.
<code>mean</code>	( <code>hmmNorm</code> , <code>hmmLNorm</code> , <code>hmmTNorm</code> ) Mean defining a Normal, or truncated Normal distribution.
<code>sd</code>	( <code>hmmNorm</code> , <code>hmmLNorm</code> , <code>hmmTNorm</code> ) Standard deviation defining a Normal, or truncated Normal distribution.
<code>meanlog</code>	( <code>hmmNorm</code> , <code>hmmLNorm</code> , <code>hmmTNorm</code> ) Mean on the log scale, for a log Normal distribution.
<code>sdlog</code>	( <code>hmmNorm</code> , <code>hmmLNorm</code> , <code>hmmTNorm</code> ) Standard deviation on the log scale, for a log Normal distribution.
<code>rate</code>	( <code>hmmPois</code> , <code>hmmExp</code> , <code>hmmGamma</code> ) Rate of a Poisson, Exponential or Gamma distribution (see <code>dpois</code> , <code>dexp</code> , <code>dgamma</code> ).
<code>shape</code>	( <code>hmmPois</code> , <code>hmmExp</code> , <code>hmmGamma</code> ) Shape parameter of a Gamma or Weibull distribution (see <code>dgamma</code> , <code>dweibull</code> ).
<code>scale</code>	( <code>hmmGamma</code> ) Shape parameter of a Gamma distribution (see <code>dgamma</code> ).
<code>size</code>	Order of a Binomial distribution (see <code>dbinom</code> ).
<code>disp</code>	Dispersion parameter of a negative binomial distribution, also called <code>size</code> or <code>order</code> . (see <code>dnbinom</code> ).
<code>lower</code>	( <code>hmmUnif</code> , <code>hmmTNorm</code> , <code>hmmMEUnif</code> ) Lower limit for an Uniform or truncated Normal distribution.
<code>upper</code>	( <code>hmmUnif</code> , <code>hmmTNorm</code> , <code>hmmMEUnif</code> ) Upper limit for an Uniform or truncated Normal distribution.
<code>sderr</code>	( <code>hmmMETNorm</code> , <code>hmmUnif</code> ) Standard deviation of the Normal measurement error distribution.
<code>meanerr</code>	( <code>hmmMETNorm</code> , <code>hmmUnif</code> ) Additional shift in the measurement error, fixed to 0 by default. This may be modelled in terms of covariates.

## Details

See the PDF manual ‘msm-manual.pdf’ in the ‘doc’ subdirectory for algebraic definitions of all these distributions.

Parameters which can be modelled in terms of covariates, on the scale of a link function, are as follows.

PARAMETER NAME	LINK FUNCTION
<code>mean</code>	identity
<code>meanlog</code>	identity
<code>rate</code>	log
<code>scale</code>	log
<code>meanerr</code>	identity
<code>prob</code>	logit

Parameters `basecat`, `lower`, `upper`, `size`, `meanerr` are fixed at their initial values. All other parameters are estimated while fitting the hidden Markov model, unless the appropriate `fixedpars` argument is supplied to `msm`.



For categorical response distributions (`hmmCat`) the outcome probabilities initialized to zero are fixed at zero, and the probability corresponding to `basecat` is fixed to one minus the sum of the remaining probabilities. These remaining probabilities are estimated, and can be modelled in terms of covariates.

### Value

Each function returns an object of class `hmodel`, which is a list containing information about the model. The only component which may be useful to end users is `r`, a function of one argument `n` which returns a random sample of size `n` from the given distribution.

### Author(s)

C. H. Jackson ([chris.jackson@imperial.ac.uk](mailto:chris.jackson@imperial.ac.uk))

### References

Satten, G.A. and Longini, I.M. Markov chains with measurement error: estimating the 'true' course of a marker of the progression of human immunodeficiency virus disease (with discussion) *Applied Statistics* 45(3): 275-309 (1996).

Jackson, C.H. and Sharples, L.D. Hidden Markov models for the onset and progresison of bronchiolitis obliterans syndrome in lung transplant recipients *Statistics in Medicine*, 21(1): 113-128 (2002).

Jackson, C.H., Sharples, L.D., Thompson, S.G. and Duffy, S.W. and Couto, E. Multi-state Markov models for disease progression with classification error. *The Statistician*, 52(2): 193-209 (2003).

### See Also

[msm](#)

---

logLik.msm	<i>Extract model log-likelihood</i>
------------	-------------------------------------

---

### Description

Extract the log-likelihood and the number of parameters of a model fitted with [msm](#).

### Usage

```
## S3 method for class 'msm':
logLik(object, ...)
```

### Arguments

<code>object</code>	A fitted multi-state model object, as returned by <a href="#">msm</a> .
<code>...</code>	(unused) further arguments passed to or from other methods.

**Value**

The minus log-likelihood of the model represented by 'object' evaluated at the maximum likelihood estimates.

**Author(s)**

C. H. Jackson <chris.jackson@imperial.ac.uk>

**See Also**

[msm](#)

---

medists

*Measurement error distributions*

---

**Description**

Truncated Normal and Uniform distributions, where the response is also subject to a Normally distributed measurement error.

**Usage**

```
dmenorm(x, mean=0, sd=1, lower=-Inf, upper=Inf, sderr=0, meanerr=0, log = FALSE)
pmenorm(q, mean=0, sd=1, lower=-Inf, upper=Inf, sderr=0, meanerr=0,
        lower.tail = TRUE, log.p = FALSE)
qmenorm(p, mean=0, sd=1, lower=-Inf, upper=Inf, sderr=0, meanerr=0,
        lower.tail = TRUE, log.p = FALSE)
rmenorm(n, mean=0, sd=1, lower=-Inf, upper=Inf, sderr=0, meanerr=0)
dmeunif(x, lower=0, upper=1, sderr=0, meanerr=0, log = FALSE)
pmeunif(q, lower=0, upper=1, sderr=0, meanerr=0, lower.tail = TRUE, log.p = FALSE)
qmeunif(p, lower=0, upper=1, sderr=0, meanerr=0, lower.tail = TRUE, log.p = FALSE)
rmeunif(n, lower=0, upper=1, sderr=0, meanerr=0)
```

**Arguments**

<code>x, q</code>	vector of quantiles.
<code>p</code>	vector of probabilities.
<code>n</code>	number of observations. If <code>length(n) &gt; 1</code> , the length is taken to be the number required.
<code>mean</code>	vector of means.
<code>sd</code>	vector of standard deviations.
<code>lower</code>	lower truncation point.
<code>upper</code>	upper truncation point.
<code>sderr</code>	Standard deviation of measurement error distribution.
<code>meanerr</code>	Optional shift for the measurement error distribution.
<code>log, log.p</code>	logical; if TRUE, probabilities $p$ are given as $\log(p)$ .
<code>lower.tail</code>	logical; if TRUE (default), probabilities are $P[X \leq x]$ , otherwise, $P[X > x]$ .

## Details

The normal distribution with measurement error has density

$$\frac{\Phi(u, \mu_2, \sigma_3) - \Phi(l, \mu_2, \sigma_3)}{\Phi(u, \mu_0, \sigma_0) - \Phi(l, \mu_0, \sigma_0)} \phi(x, \mu_0 + \mu_\epsilon, \sigma_2)$$

where

$$\begin{aligned}\sigma_2^2 &= \sigma_0^2 + \sigma_\epsilon^2, \\ \sigma_3 &= \sigma_0 \sigma_\epsilon / \sigma_2, \\ \mu_2 &= (x - \mu_\epsilon) \sigma_0^2 + \mu_0 \sigma_\epsilon^2,\end{aligned}$$

$\mu_0$  is the mean of the original Normal distribution before truncation,

$\sigma_0$  is the corresponding standard deviation,

$u$  is the upper truncation point,

$l$  is the lower truncation point,

$\sigma_\epsilon$  is the standard deviation of the additional measurement error,

$\mu_\epsilon$  is the mean of the measurement error (usually 0).

$\phi(x)$  is the density of the corresponding normal distribution, and

$\Phi(x)$  is the distribution function of the corresponding normal distribution.

The uniform distribution with measurement error has density

$$(\Phi(x, \mu_\epsilon + l, \sigma_\epsilon) - \Phi(x, \mu_\epsilon + u, \sigma_\epsilon)) / (u - l)$$

These are calculated from the original truncated Normal or Uniform density functions  $f(\cdot | \mu, \sigma, l, u)$  as

$$\int f(y | \mu, \sigma, l, u) \phi(x, y + \mu_\epsilon, \sigma_\epsilon) dy$$

If `sderr` and `meanerr` are not specified they assume the default values of 0, representing no measurement error variance, and no constant shift in the measurement error, respectively.

Therefore, for example with no other arguments, `dmenorm(x)`, is simply equivalent to `dtnorm(x)`, which in turn is equivalent to `dnorm(x)`.

These distributions were used by Satten and Longini (1996) for CD4 cell counts conditionally on hidden Markov states of HIV infection, and later by Jackson and Sharples (2002) for FEV1 measurements conditionally on states of chronic lung transplant rejection.

These distribution functions are just provided for convenience, and are not optimised for numerical accuracy. To fit a hidden Markov model with these response distributions, use a `hmmMETNorm` or `hmmMEUnif` constructor. See the `hmm-dists` help page for further details.

## Value

`dmenorm`, `dmeunif` give the density, `pmenorm`, `pmeunif` give the distribution function, `qmenorm`, `qmeunif` give the quantile function, and `rmenorm`, `rmeunif` generate random deviates, for the Normal and Uniform versions respectively.

**Author(s)**

C. H. Jackson <chris.jackson@imperial.ac.uk>

**References**

Satten, G.A. and Longini, I.M. Markov chains with measurement error: estimating the 'true' course of a marker of the progression of human immunodeficiency virus disease (with discussion) *Applied Statistics* 45(3): 275-309 (1996)

Jackson, C.H. and Sharples, L.D. Hidden Markov models for the onset and progression of bronchiolitis obliterans syndrome in lung transplant recipients *Statistics in Medicine*, 21(1): 113–128 (2002).

**See Also**

[dnorm](#), [dunif](#), [dtnorm](#)

**Examples**

```
## what does the distribution look like?
x <- seq(50, 90, by=1)
plot(x, dnorm(x, 70, 10), type="l", ylim=c(0,0.06)) ## standard Normal
lines(x, dtnorm(x, 70, 10, 60, 80), type="l")      ## truncated Normal
## truncated Normal with small measurement error
lines(x, dmenorm(x, 70, 10, 60, 80, sderr=3), type="l")
```

---

msm

---

*Multi-state Markov and hidden Markov models in continuous time*


---

**Description**

Fit a continuous-time Markov or hidden Markov multi-state model by maximum likelihood. Observations of the process can be made at arbitrary times, or the exact times of transition between states can be known. Covariates can be fitted to the Markov chain transition intensities or to the hidden Markov observation process.

**Usage**

```
msm ( formula, subject=NULL, data = list(), qmatrix, gen.inits = FALSE,
      ematrix=NULL, hmodel=NULL, obstype=NULL,
      covariates = NULL, covinits = NULL, constraint = NULL,
      misccovariates = NULL, misccovinits = NULL, miscconstraint = NULL,
      hcovariates = NULL, hcovinits = NULL, hconstraint = NULL,
      qconstraint=NULL, econstraint=NULL, initprobs = NULL, est.initprobs=FALSE,
      death = FALSE, exacttimes = FALSE, censor=NULL,
      censor.states=NULL, cl = 0.95, fixedpars = NULL, center=TRUE,
      opt.method=c("optim","nlm"), hessian=TRUE, use.deriv=FALSE,
      deriv.test=FALSE, analyticp=TRUE, ... )
```

**Arguments**

- `formula` A formula giving the vectors containing the observed states and the corresponding observation times. For example,  

```
state ~ time
```

Observed states should be in the set  $1, \dots, n$ , where  $n$  is the number of states.
- `subject` Vector of subject identification numbers for the data specified by `formula`. If missing, then all observations are assumed to be on the same subject. These must be sorted so that all observations on the same subject are adjacent.
- `data` Optional data frame in which to interpret the variables supplied in `formula`, `subject`, `covariates`, `misccovariates`, `hcovariates` and `obstype`.
- `qmatrix` Initial transition intensity matrix of the Markov chain. If an instantaneous transition is not allowed from state  $r$  to state  $s$ , then `qmatrix` should have  $(r, s)$  entry 0, otherwise it should be non-zero. Any diagonal entry of `qmatrix` is ignored, as it is constrained to be equal to minus the sum of the rest of the row. For example,  

```
rbind( c( 0, 0.1, 0.01 ), c( 0.1, 0, 0.2 ), c( 0, 0, 0 ) )
```

represents a 'health - disease - death' model, with transition intensities 0.1 from health to disease, 0.01 from health to death, 0.1 from disease to health, and 0.2 from disease to death. The initial intensities given here are with any covariates set to their means in the data (or set to zero, if `center = FALSE`).
- `gen.inits` If TRUE, then initial values for the transition intensities are estimated by assuming that the data represent the exact transition times of the process. The non-zero entries of the supplied `qmatrix` are assumed to indicate the allowed transitions of the model.
- `ematrix` If misclassification between states is to be modelled, this should be a matrix of initial values for the misclassification probabilities. The rows represent underlying states, and the columns represent observed states. If an observation of state  $s$  is not possible when the subject occupies underlying state  $r$ , then `ematrix` should have  $(r, s)$  entry 0. Otherwise `ematrix` should have  $(r, s)$  entry corresponding to the probability of observing  $s$  conditionally on occupying true state  $r$ . The diagonal of `ematrix` is ignored, as rows are constrained to sum to 1. For example,  

```
rbind( c( 0, 0.1, 0 ), c( 0.1, 0, 0.1 ), c( 0, 0.1, 0 ) )
```

represents a model in which misclassifications are only permitted between adjacent states.
- `hmodel` For an alternative way of specifying misclassification models, see `hmodel`. Specification of the hidden Markov model. This should be a list of return values from the constructor functions described in the [hmm-dists](#) help page. Each

element of the list corresponds to the outcome model conditionally on the corresponding underlying state.

For example, consider a three-state hidden Markov model. Suppose the observations in underlying state 1 are generated from a Normal distribution with mean 100 and standard deviation 16, while observations in underlying state 2 are Normal with mean 54 and standard deviation 18. Observations in state 3, representing death, are exactly observed, and coded as 999 in the data. This model is specified as

```
hmodel = list(hmmNorm(mean=100, sd=16), hmmNorm(mean=54,
sd=18), hmmIdent(999))
```

The mean and standard deviation parameters are estimated starting from these initial values. See the `hmm-dists` help page for details of the constructor functions for each available distribution.

A misclassification model, that is, a hidden Markov model where the outcomes are misclassified observations of the underlying states, can either be specified using a list of `hmmCat` objects, or by using an `ematrix` as in previous versions of **msm**.

For example,

```
ematrix = rbind( c( 0, 0.1, 0, 0 ), c( 0.1, 0, 0.1, 0
), c( 0, 0.1, 0, 0 ), c( 0, 0, 0, 0 ) )
```

is equivalent to

```
hmodel = list( hmmCat(prob=c(0.9, 0.1, 0, 0)), hmmCat(prob=c(0.1,
0.8, 0.1, 0)), hmmCat(prob=c(0, 0.1, 0.9, 0)), hmmIdent())
```

`obstype`

A vector specifying the observation scheme for each row of the data. This can be included in the data frame `data` along with the state, time, subject IDs and covariates. Its elements should be either 1, 2 or 3, meaning as follows:

- 1** An observation of the process at an arbitrary time (a "snapshot" of the process)
- 2** An exact transition time, with the state at the previous observation retained until the current observation.
- 3** An exact transition time, but the state at the instant before entering this state is unknown. A common example is death times in studies of chronic diseases.

If `obstype` is not specified, this defaults to all 1. If `obstype` is a single number, all observations are assumed to be of this type.

This is a generalisation of the `death` and `exacttimes` arguments to allow different schemes per observation.

`exacttimes=TRUE` specifies that all observations are of `obstype 2`.

`death = death.states` specifies that all observations of `death.states` are of type 3. `death = TRUE` specifies that all observations in the final absorbing state are of type 3.

`covariates`

Formula representing the covariates on the transition intensities via a log-linear model. For example,

```
~ age + sex + treatment
```

- covinits** Initial values for log-linear effects of covariates on the transition intensities. This should be a named list with each element corresponding to a covariate. A single element contains the initial values for that covariate on each transition intensity, reading across the rows in order. For a pair of effects constrained to be equal, the initial value for the first of the two effects is used.
- For example, for a model with the above `qmatrix` and age and sex covariates, the following initialises all covariate effects to zero apart from the age effect on the 2-1 transition, and the sex effect on the 1-3 transition. `covinits = list(sex=c(0, 0, 0.1, 0), age=c(0, 0.1, 0, 0))`
- For factor covariates, name each level by concatenating the name of the covariate with the level name, quoting if necessary. For example, for a covariate `agegroup` with three levels 0-15, 15-60, 60-, use something like `covinits = list("agegroup15-60"=c(0, 0.1, 0, 0), "agegroup60-"=c(0.1, 0.1, 0, 0))`
- If not specified or wrongly specified, initial values are assumed to be zero.
- constraint** A list of one vector for each named covariate. The vector indicates which covariate effects on intensities are constrained to be equal. Take, for example, a model with five transition intensities and two covariates. Specifying `constraint = list (age = c(1,1,1,2,2), treatment = c(1,2,3,4,5))`
- constrains the effect of age to be equal for the first three intensities, and equal for the fourth and fifth. The effect of treatment is assumed to be different for each intensity. Any vector of increasing numbers can be used as indicators. The intensity parameters are assumed to be ordered by reading across the rows of the transition matrix, starting at the first row, ignoring the diagonals.
- For categorical covariates, defined using `factor(covname)`, specify constraints as follows:
- ```
list(..., covnameVALUE1 = c(...), covnameVALUE2 = c(...), ...)
```
- where `VALUE1, VALUE2, ...` are the levels of the factor. Make sure the `contrasts` option is set appropriately, for example, the default `options(contrasts=c(contr.treatment, contr.poly))` sets the first (baseline) level of unordered factors to zero.
- To assume no covariate effect on a certain transition, set its initial value to zero and use the `fixedpars` argument to fix it during the optimisation.
- misccovariates** A formula representing the covariates on the misclassification probabilities, analogously to `covariates`. Only used if the model is specified using `ematrix`, rather than `hmodel`.
- misccovinits** Initial values for the covariates on the misclassification probabilities, defined in the same way as `covinits`. Only used if the model is specified using `ematrix`.

|                            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|----------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>misconstraint</code> | A list of one vector for each named covariate on misclassification probabilities. The vector indicates which covariate effects on misclassification probabilities are constrained to be equal, analogously to <code>constraint</code> . Only used if the model is specified using <code>ematrix</code> .                                                                                                                                                                                                                                                                                                                                             |
| <code>hcovariates</code>   | List of formulae the same length as <code>hmodel</code> , defining any covariates governing the hidden Markov outcome models. The covariates operate on a suitably link-transformed linear scale, for example, log scale for a Poisson outcome model. If there are no covariates for a certain hidden state, then insert a <code>NULL</code> in the corresponding place in the list. For example, <code>hcovariates = list(~acute + age, ~acute, NULL)</code> .                                                                                                                                                                                      |
| <code>hcovinits</code>     | Initial values for the hidden Markov model covariates. A list of the same length as <code>hcovariates</code> . Each element is a vector with initial values for each covariate on that state. For example, the above <code>hcovariates</code> can be initialised with <code>hcovariates = list(c(-8, 0), -8, NULL)</code> . Initial values must be given for all or no covariates, if none are given these are all set to zero. The initial value given in the <code>hmodel</code> constructor function for the corresponding baseline parameter is interpreted as the value of that parameter with any covariates fixed to their means in the data. |
| <code>hconstraint</code>   | A named list. Each element is a vector of constraints on the named hidden Markov model parameter. The vector has length equal to the number of times that class of parameter appears in the whole model.<br>For example consider the three-state hidden Markov model described above, with normally-distributed outcomes for states 1 and 2. To constrain the outcome variance to be equal for states 1 and 2, and to also constrain the effect of <code>acute</code> on the outcome mean to be equal for states 1 and 2, specify<br><code>hconstraint = list(sd = c(1,1), acute=c(1,1))</code>                                                      |
| <code>qconstraint</code>   | A vector of indicators specifying which baseline transition intensities are equal. For example,<br><code>qconstraint = c(1,2,3,3)</code><br>constrains the third and fourth intensities to be equal, in a model with four allowed instantaneous transitions.                                                                                                                                                                                                                                                                                                                                                                                         |
| <code>econstraint</code>   | A similar vector of indicators specifying which baseline misclassification probabilities are constrained to be equal. Only used if the model is specified using <code>ematrix</code> , rather than <code>hmodel</code> .                                                                                                                                                                                                                                                                                                                                                                                                                             |
| <code>initprobs</code>     | Currently only used in hidden Markov models. Vector of assumed underlying state occupancy probabilities at each individual's first observation. If these are estimated (see <code>est.initprobs</code> ), then this defaults to equal probability for each state. Otherwise this defaults to <code>c(1, rep(0, nstates-1))</code> , that is, in state 1 with a probability of 1. Scaled to sum to 1 if necessary.                                                                                                                                                                                                                                    |
| <code>est.initprobs</code> | If <code>TRUE</code> , then the underlying state occupancy probabilities at the first observation will be estimated, starting from initial values taken from the <code>initprobs</code> argument. Be warned that if any of these initial values are 0 or 1, then <code>optim</code> will give a "non-finite value" error. To fix any of these probabilities during the estimation, e.g. at values of 0 or 1, then use an appropriate <code>fixedpars</code> argument. Note that the free parameters during this estimation excludes the state 1                                                                                                      |



|               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|               | occupancy probability, which is fixed at 1 minus the sum of the other probabilities.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| death         | <p>Vector of indices of the death states. A death state is an absorbing state whose time of entry is known exactly, but the individual is assumed to be in an unknown transient state ("alive") at the previous instant. This is the usual situation for times of death in chronic disease monitoring data. For example, if you specify <code>death = c(4, 5)</code> then states 4 and 5 are assumed to be death states.</p> <p><code>death = TRUE</code> indicates that the final state is a death state, and <code>death = FALSE</code> (the default) indicates that there is no death state. See the <code>obstype</code> argument.</p>                                                                                                                                                                                                                                                                                                                                  |
| censor        | <p>A state, or vector of states, which indicates censoring. Censoring means that the observed state is known only to be one of a particular set of states. For example, <code>censor=999</code> indicates that all observations of 999 in the vector of observed states denote censoring times. By default, this means that the true state could have been anything other than an absorbing state. To specify corresponding true states explicitly, use a <code>censor.states</code> argument.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| censor.states | <p>Specifies the underlying states which censored observations can represent. If <code>censor</code> is a single number (the default) this can be a vector, or a list with one element. If <code>censor</code> is a vector with more than one element, this should be a list, with each element a vector corresponding to the equivalent element of <code>censor</code>. For example</p> <pre>censor = c(99, 999), censor.states = list(c(2,3), c(3,4))</pre> <p>means that observations coded 99 represent either state 2 or state 3, while observations coded 999 are really either state 3 or state 4.</p>                                                                                                                                                                                                                                                                                                                                                               |
| exacttimes    | <p>By default, the transitions of the Markov process are assumed to take place at unknown occasions in between the observation times. If <code>exacttimes</code> is set to <code>TRUE</code>, then all observation times are assumed to represent the exact and complete times of transition of the process. This is equivalent to every row of the data having <code>obstype = 2</code>. See the <code>obstype</code> argument.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| cl            | <p>Width of symmetric confidence intervals for maximum likelihood estimates, by default 0.95.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| fixedpars     | <p>Vector of indices of parameters whose values will be fixed at their initial values during the optimisation. These are given in the order: transition intensities (reading across rows of the transition matrix), covariates on intensities (ordered by intensities within covariates), hidden Markov model parameters (ordered by parameters within states), hidden Markov model covariate parameters (ordered by covariates within parameters within states), initial state occupancy probabilities (excluding the first probability, which is fixed at one minus the sum of the others).</p> <p>For covariates on misclassification probabilities, this is a change from version 0.4 in the parameter ordering. Previously these were ordered by misclassification probabilities within covariates.</p> <p>This can be useful for profiling likelihoods, and building complex models stage by stage. To fix all parameters, specify <code>fixedpars = TRUE</code>.</p> |
| center        | <p>If <code>TRUE</code> (the default) then covariates are centered at their means during the maximum likelihood estimation. This usually improves convergence.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |

|                         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|-------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>opt.method</code> | Quoted name of the R function to perform minimisation of the minus twice log likelihood. Either "optim" or "nlm". <code>optim</code> is the default.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| <code>hessian</code>    | If TRUE (the default) then the Hessian matrix is computed at the maximum likelihood estimates, to obtain standard errors and confidence intervals.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| <code>use.deriv</code>  | If TRUE then analytic first derivatives are used in the optimisation of the likelihood, when an appropriate quasi-Newton optimisation method, such as BFGS, is being used. Note that the default for <code>optim</code> is a Nelder-Mead method which cannot use derivatives. However, these derivatives, if supplied, are always used to calculate the Hessian.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| <code>deriv.test</code> | If TRUE, then analytic and numeric derivatives are computed and compared at the initial values, and no optimisation is performed.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| <code>analyticp</code>  | By default, the likelihood for certain simpler 3, 4 and 5 state models is calculated using an analytic expression for the transition probability (P) matrix. To revert to the original method of using the matrix exponential, specify <code>analyticp=FALSE</code> . See the PDF manual for a list of the models for which analytic P matrices are implemented.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| <code>...</code>        | Optional arguments to the general-purpose R optimisation routines, <code>optim</code> or <code>nlm</code> . Useful options for <code>optim</code> include <code>method="BFGS"</code> for using a quasi-Newton optimisation algorithm, which can often be faster than the default Nelder-Mead. If the optimisation fails to converge, consider normalising the problem using, for example, <code>control=list(fnscale = 2500)</code> , for example, replacing 2500 by a number of the order of magnitude of the likelihood. If 'false' convergence is reported and the standard errors cannot be calculated due to a non-positive-definite Hessian, then consider tightening the tolerance criteria for convergence. If the optimisation takes a long time, intermediate steps can be printed using the <code>trace</code> argument of the control list. See <code>optim</code> for details. |

## Details

For full details about the methodology behind the **msm** package, refer to the PDF manual 'msm-manual.pdf' in the 'doc' subdirectory of the package. This includes a tutorial in the typical use of **msm**.

Users upgrading from versions of **msm** less than 0.5 will need to change some of their model fitting syntax. In particular, initial values are now specified in the `qmatrix` and `covinits` arguments instead of `inits`, and `qmatrix` is no longer a matrix of 0/1 indicators. See the appendix to the PDF manual or the NEWS file in the top-level installation directory for a full list of changes.

For simple multi-state Markov models, the likelihood is calculated in terms of the transition intensity matrix  $Q$ . When the data consist of observations of the Markov process at arbitrary times, the exact transition times are not known. Then the likelihood is calculated using the transition probability matrix  $P(t) = \exp(tQ)$ , where  $\exp$  is the matrix exponential. If state  $i$  is observed at time  $t$  and state  $j$  is observed at time  $u$ , then the contribution to the likelihood from this pair of observations is the  $i, j$  element of  $P(u - t)$ . See, for example, Kalbfleisch and Lawless (1985), Kay (1986), or Gentleman *et al.* (1994).

For hidden Markov models, the likelihood for an individual with  $k$  observations is calculated directly by summing over the unknown state at each time, producing a product of  $k$  matrices. The

calculation is a generalisation of the method described by Satten and Longini (1996), and also by Jackson and Sharples (2002), and Jackson *et al.* (2003).

There must be enough information in the data on each state to estimate each transition rate, otherwise the likelihood will be flat and the maximum will not be found. It may be appropriate to reduce the number of states in the model, or reduce the number of covariate effects, to ensure convergence. Hidden Markov models are particularly susceptible to non-identifiability, especially when combined with a complex transition matrix.

Choosing an appropriate set of initial values for the optimisation can also be important. For flat likelihoods, 'informative' initial values will often be required.

## Value

A list of class `msm`, with components:

|                                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|-------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>call</code>                   | The original call to <code>msm</code> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| <code>Qmatrices</code>              | A list of matrices. The first component, labelled <code>logbaseline</code> , is a matrix containing the estimated transition intensities on the log scale with any covariates fixed at their means in the data. Each remaining component is a matrix giving the linear effects of the labelled covariate on the matrix of log intensities. To extract an estimated intensity matrix on the natural scale, at an arbitrary combination of covariate values, use the function <code>qmatrix.msm</code> .                      |
| <code>QmatricesSE</code>            | The standard error matrices corresponding to <code>Qmatrices</code> .                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| <code>QmatricesL, QmatricesU</code> | Corresponding lower and upper symmetric confidence limits, of width 0.95 unless specified otherwise by the <code>cl</code> argument.                                                                                                                                                                                                                                                                                                                                                                                        |
| <code>Ematrices</code>              | A list of matrices. The first component, labelled <code>logitbaseline</code> , is the estimated misclassification probability matrix with any covariates fixed at their means in the data. Each remaining component is a matrix giving the linear effects of the labelled covariate on the matrix of logit misclassification probabilities. To extract an estimated misclassification probability matrix on the natural scale, at an arbitrary combination of covariate values, use the function <code>ematrix.msm</code> . |
| <code>EmatricesSE</code>            | The standard error matrices corresponding to <code>Ematrices</code> .                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| <code>EmatricesL, EmatricesU</code> | Corresponding lower and upper symmetric confidence limits, of width 0.95 unless specified otherwise by the <code>cl</code> argument.                                                                                                                                                                                                                                                                                                                                                                                        |
| <code>sojourn</code>                | A list with components:<br><code>mean</code> = estimated mean sojourn times in the transient states, with covariates fixed at their means.<br><code>se</code> = corresponding standard errors.                                                                                                                                                                                                                                                                                                                              |
| <code>minus2loglik</code>           | Minus twice the maximised log-likelihood.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| <code>estimates</code>              | Vector of untransformed maximum likelihood estimates returned from <code>optim</code> . Transition intensities are on the log scale and misclassification probabilities are the logit scale.                                                                                                                                                                                                                                                                                                                                |
| <code>estimates.t</code>            | Vector of transformed maximum likelihood estimates with intensities and probabilities on their natural scales.                                                                                                                                                                                                                                                                                                                                                                                                              |

|                        |                                                                                                                                                                                                                                                        |
|------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>fixedpars</code> | Indices of <code>estimates</code> which were fixed during the maximum likelihood estimation.                                                                                                                                                           |
| <code>covmat</code>    | Covariance matrix corresponding to <code>estimates</code> .                                                                                                                                                                                            |
| <code>ci</code>        | Matrix of confidence intervals corresponding to <code>estimates</code>                                                                                                                                                                                 |
| <code>opt</code>       | Return value from <code>optim</code> or <code>nlm</code> , giving information about the results of the optimisation.                                                                                                                                   |
| <code>foundse</code>   | Logical value indicating whether the Hessian was positive-definite at the supposed maximum of the likelihood. If not, the covariance matrix of the parameters is unavailable. In these cases the optimisation has probably not converged to a maximum. |
| <code>data</code>      | A list of constants and vectors giving the data, for use in post-processing.                                                                                                                                                                           |
| <code>qmodel</code>    | A list of objects specifying the model for transition intensities, for use in post-processing.                                                                                                                                                         |
| <code>emodel</code>    | A list of objects specifying the model for misclassification.                                                                                                                                                                                          |
| <code>qcmmodel</code>  | A list of objects specifying the model for covariates on the transition intensities.                                                                                                                                                                   |
| <code>ecmodel</code>   | A list of objects specifying the model for covariates on misclassification probabilities.                                                                                                                                                              |
| <code>hmodel</code>    | A list of class "hmodel", containing objects specifying the hidden Markov model. Estimates of "baseline" location parameters are presented with any covariates fixed to their means in the data.                                                       |
| <code>cmodel</code>    | A list of objects specifying any model for censoring.                                                                                                                                                                                                  |

Printing a `msm` object by typing the object's name at the command line implicitly invokes `print.msm`. This formats and prints the important information in the model fit. This includes the fitted transition intensity matrix, matrices containing covariate effects on intensities, and mean sojourn times from a fitted `msm` model. When there is a hidden Markov model, the chief information in the `hmodel` component is also formatted and printed. This includes estimates and confidence intervals for each parameter.

To extract summary information from the fitted model, it is recommended to use the more flexible extractor functions, such as `qmatrix.msm`, `pmatrix.msm`, `sojourn.msm`, instead of directly reading from list components of `msm` objects.

### Author(s)

C. H. Jackson <chris.jackson@imperial.ac.uk>

### References

- Kalbfleisch, J., Lawless, J.F., The analysis of panel data under a Markov assumption *Journal of the American Statistical Association* (1985) 80(392): 863–871.
- Kay, R. A Markov model for analysing cancer markers and disease states in survival studies. *Biometrics* (1986) 42: 855–865.
- Gentleman, R.C., Lawless, J.F., Lindsey, J.C. and Yan, P. Multi-state Markov models for analysing incomplete disease history data with illustrations for HIV disease. *Statistics in Medicine* (1994) 13(3): 805–821.

Satten, G.A. and Longini, I.M. Markov chains with measurement error: estimating the 'true' course of a marker of the progression of human immunodeficiency virus disease (with discussion) *Applied Statistics* 45(3): 275-309 (1996)

Jackson, C.H. and Sharples, L.D. Hidden Markov models for the onset and progression of bronchiolitis obliterans syndrome in lung transplant recipients *Statistics in Medicine*, 21(1): 113–128 (2002).

Jackson, C.H., Sharples, L.D., Thompson, S.G. and Duffy, S.W. and Couto, E. Multi-state Markov models for disease progression with classification error. *The Statistician*, 52(2): 193–209 (2003)

### See Also

[simmulti.msm](#), [plot.msm](#), [summary.msm](#), [qmatrix.msm](#), [pmatrix.msm](#), [sojourn.msm](#).

### Examples

```
### Heart transplant data
### For further details and background to this example, see
### the PDF manual in the doc directory.
data(heart)
print(heart[1:10,])
twoway4.q <- rbind(c(-0.5, 0.25, 0, 0.25), c(0.166, -0.498, 0.166, 0.166),
c(0, 0.25, -0.5, 0.25), c(0, 0, 0, 0))
statetable.msm(state, PTNUM, data=heart)
crudeinits.msm(state ~ years, PTNUM, data=heart, qmatrix=twoway4.q)
heart.msm <- msm( state ~ years, subject=PTNUM, data = heart,
                qmatrix = twoway4.q, death = 4,
                control = list ( trace = 2, REPORT = 1 ) )

heart.msm
qmatrix.msm(heart.msm)
pmatrix.msm(heart.msm, t=10)
sojourn.msm(heart.msm)
```

---

odds.msm

*Calculate tables of odds ratios for covariates on misclassification probabilities*

---

### Description

Odds ratios are computed by exponentiating the estimated covariate effects on the logit-misclassification probabilities.

### Usage

```
odds.msm(x, odds.scale = 1, cl = 0.95)
```

**Arguments**

|                         |                                                                                                                                                                                          |
|-------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>x</code>          | Output from <code>msm</code> representing a fitted multi-state model.                                                                                                                    |
| <code>odds.scale</code> | Vector with same elements as number of covariates on misclassification probabilities. Corresponds to the increase in each covariate used to calculate its odds ratio. Defaults to all 1. |
| <code>cl</code>         | Width of the symmetric confidence interval to present. Defaults to 0.95.                                                                                                                 |

**Value**

A list of tables containing odds ratio estimates, one table for each covariate. Each table has three columns, containing the odds ratio, and an approximate upper 95% and lower 95% confidence limit respectively (assuming normality on the log scale), for each misclassification probability.

**Author(s)**

C. H. Jackson ([chris.jackson@imperial.ac.uk](mailto:chris.jackson@imperial.ac.uk))

**See Also**

[msm](#), [hazard.msm](#)

---

pexp

*Exponential distribution with piecewise-constant rate*

---

**Description**

Density, distribution function, quantile function and random generation for a generalisation of the exponential distribution, in which the rate changes at a series of times.

**Usage**

```
dpexp(x, rate=1, t=0, log = FALSE)
ppexp(q, rate=1, t=0, lower.tail = TRUE, log.p = FALSE)
qpexp(p, rate=1, t=0, lower.tail = TRUE, log.p = FALSE)
rpexp(n, rate=1, t=0)
```

**Arguments**

|                         |                                                                                                                                                                                                 |
|-------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>x, q</code>       | vector of quantiles.                                                                                                                                                                            |
| <code>p</code>          | vector of probabilities.                                                                                                                                                                        |
| <code>n</code>          | number of observations. If <code>length(n) &gt; 1</code> , the length is taken to be the number required.                                                                                       |
| <code>rate</code>       | vector of rates.                                                                                                                                                                                |
| <code>t</code>          | vector of the same length as <code>rate</code> , giving the times at which the rate changes. The first element of <code>t</code> should be 0, and <code>t</code> should be in increasing order. |
| <code>log, log.p</code> | logical; if TRUE, probabilities <code>p</code> are given as <code>log(p)</code> .                                                                                                               |
| <code>lower.tail</code> | logical; if TRUE (default), probabilities are $P[X \leq x]$ , otherwise, $P[X > x]$ .                                                                                                           |

## Details

Consider the exponential distribution with rates  $r_1, \dots, r_n$  changing at times  $t_1, \dots, t_n$ , with  $t_1 = 0$ . Suppose  $t_k$  is the maximum  $t_i$  such that  $t_i < x$ . The density of this distribution at  $x > 0$  is  $f(x)$  for  $k = 1$ , and

$$\prod_{i=1}^k (1 - F(t_i - t_{i-1}, r_i)) f(x - t_k, r_k)$$

for  $k > 1$ .

where  $F()$  and  $f()$  are the distribution and density functions of the standard exponential distribution.

If `rate` is of length 1, this is just the standard exponential distribution. Therefore, for example, `dpexp(x)`, with no other arguments, is simply equivalent to `dexp(x)`.

Only `rpexp` is used in the `msm` package, to simulate from Markov processes with piecewise-constant intensities depending on time-dependent covariates. These functions are merely provided for completion, and are not optimized for numerical stability.

## Value

`dpexp` gives the density, `ppexp` gives the distribution function, `qpexp` gives the quantile function, and `rpexp` generates random deviates.

## Author(s)

C. H. Jackson <chris.jackson@imperial.ac.uk>

## See Also

[dexp](#), [sim.msm](#).

## Examples

```
x <- seq(0.1, 50, by=0.1)
rate <- c(0.1, 0.2, 0.05, 0.3)
t <- c(0, 10, 20, 30)
plot(x, dexp(x, 0.1), type="l") ## standard exponential distribution
lines(x, dpexp(x, rate, t), type="l", lty=2) ## distribution with piecewise constant rate
plot(x, pexp(x, 0.1), type="l") ## standard exponential distribution
lines(x, ppexp(x, rate, t), type="l", lty=2) ## distribution with piecewise constant rate
```

## Description

This produces a plot of the expected probability of survival against time, from each transient state. Survival is defined as not entering an absorbing state.

**Usage**

```
## S3 method for class 'msm':
plot(x, from, to, range, covariates, legend.pos, ...)
```

**Arguments**

|            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| x          | Output from <code>msm</code> , representing a fitted multi-state model object                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| from       | States from which to consider survival. Defaults to the complete set of transient states.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| to         | Absorbing state to consider. Defaults to the highest-labelled absorbing state.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| range      | Vector of two elements, giving the range of times to plot for.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| covariates | Covariate values for which to evaluate the expected probabilities. This can either be: <ul style="list-style-type: none"> <li>the string "mean", denoting the means of the covariates in the data (this is the default),</li> <li>the number 0, indicating that all the covariates should be set to zero,</li> <li>or a list of values, with optional names. For example <pre>list (60, 1)</pre>           where the order of the list follows the order of the covariates originally given in the model formula, or a named list, <pre>list (age = 60, sex = 1)</pre> </li> </ul> |
| legend.pos | Vector of the <i>x</i> and <i>y</i> position, respectively, of the legend.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| ...        | Other arguments to the generic <code>plot</code> function                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |

**Author(s)**

C. H. Jackson <chris.jackson@imperial.ac.uk>

**See Also**

[msm](#)

---

pmatrix.msm

*Transition probability matrix*

---

**Description**

Extract the estimated transition probability matrix from a fitted multi-state model for a given time interval, at a given set of covariate values.



**Usage**

```
pmatrix.msm(x, t=1, covariates="mean", ci.boot=FALSE, cl=0.95, B=500)
```

**Arguments**

|            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| x          | A fitted multi-state model, as returned by <a href="#">msm</a> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| t          | The time interval to estimate the transition probabilities for, by default one unit.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| covariates | The covariate values at which to estimate the transition probabilities. This can either be: <ul style="list-style-type: none"> <li>the string "mean", denoting the means of the covariates in the data (this is the default),</li> <li>the number 0, indicating that all the covariates should be set to zero,</li> <li>or a list of values, with optional names. For example <pre>list (60, 1)</pre> where the order of the list follows the order of the covariates originally given in the model formula, or a named list, <pre>list (age = 60, sex = 1)</pre> </li> </ul> |
| ci.boot    | Calculate a bootstrap confidence interval. This is usually time-consuming, and disabled by default. See <a href="#">boot.msm</a> for more details of bootstrapping in <a href="#">msm</a> .                                                                                                                                                                                                                                                                                                                                                                                   |
| cl         | Width of the symmetric confidence interval                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| B          | Number of bootstrap replicates                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |

**Details**

For a continuous-time homogeneous Markov process with transition intensity matrix  $Q$ , the probability of occupying state  $s$  at time  $u + t$  conditionally on occupying state  $r$  at time  $u$  is given by the  $(r, s)$  entry of the matrix  $P(t) = \exp(tQ)$ .

For non-homogeneous processes, where covariates and hence the transition intensity matrix are time-dependent, but are piecewise-constant within the time interval  $[u, u+t]$ , the function [pmatrix.piecewise.msm](#) can be used.

**Value**

The matrix of estimated transition probabilities  $P(t)$  in the given time. Rows correspond to "from-state" and columns to "to-state".

**Author(s)**

C. H. Jackson <[chris.jackson@imperial.ac.uk](mailto:chris.jackson@imperial.ac.uk)>

**See Also**

[qmatrix.msm](#), [pmatrix.piecewise.msm](#), [boot.msm](#)

---

```
pmatrix.piecewise.msm
```

*Transition probability matrix for processes with piecewise-constant intensities*

---

## Description

Extract the estimated transition probability matrix from a fitted non-time-homogeneous multi-state model for a given time interval. This is a generalisation of `pmatrix.msm` to non-homogeneous models with time-dependent covariates.

## Usage

```
pmatrix.piecewise.msm(x, t1, t2, times, covariates)
```

## Arguments

|                         |                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|-------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>x</code>          | A fitted multi-state model, as returned by <code>msm</code> . This should be a non-homogeneous model, whose transition intensity matrix depends on a time-dependent covariate.                                                                                                                                                                                                                                                                            |
| <code>t1</code>         | The start of the time interval to estimate the transition probabilities for.                                                                                                                                                                                                                                                                                                                                                                              |
| <code>t2</code>         | The end of the time interval to estimate the transition probabilities for.                                                                                                                                                                                                                                                                                                                                                                                |
| <code>times</code>      | Cut points at which the transition intensity matrix changes.                                                                                                                                                                                                                                                                                                                                                                                              |
| <code>covariates</code> | A list with number of components one greater than the length of <code>times</code> . Each component of the list is specified in the same way as the <code>covariates</code> argument to <code>pmatrix.msm</code> . The components correspond to the covariate values in the intervals<br>$(t1, times[1]), (times[1], times[2]), \dots, (times[length(times)], t2)$<br>(assuming that all elements of <code>times</code> are in the interval $(t1, t2)$ ). |

## Details

Suppose a multi-state model has been fitted, in which the transition intensity matrix  $Q(x(t))$  is modelled in terms of time-dependent covariates  $x(t)$ . The transition probability matrix  $P(t_1, t_n)$  for the time interval  $(t_1, t_n)$  cannot be calculated from the estimated intensity matrix as  $\exp((t_n - t_1)Q)$ , because  $Q$  varies within the interval  $t_1, t_n$ . However, if the covariates are piecewise-constant, or can be approximated as piecewise-constant, then we can calculate  $P(t_1, t_n)$  by multiplying together individual matrices  $P(t_i, t_{i+1}) = \exp((t_{i+1} - t_i)Q)$ , calculated over intervals where  $Q$  is constant:

$$P(t_1, t_n) = P(t_1, t_2)P(t_2, t_3) \dots P(t_{n-1}, t_n)$$

## Value

The matrix of estimated transition probabilities  $P(t)$  for the time interval  $[t1, tn]$ . That is, the probabilities of occupying state  $s$  at time  $t_n$  conditionally on occupying state  $r$  at time  $t_1$ . Rows correspond to "from-state" and columns to "to-state".

**Author(s)**

C. H. Jackson <chris.jackson@imperial.ac.uk>

**See Also**

[pmatrix.msm](#)

**Examples**

```
## Not run:
## In a clinical study, suppose patients are given a placebo in the
## first 5 weeks, then they begin treatment 1 at 5 weeks, and
## a combination of treatments 1 and 2 from 10 weeks.
## Suppose a multi-state model x has been fitted for the patients'
## progress, with treat1 and treat2 as time dependent covariates.

## Cut points for when treatment covariate changes
times <- c(0, 5, 10)

## Indicators for which treatments are active at the three cut points
covariates <- list( list (treat1=0, treat2=0), list(treat1=1, treat2=0),
list(treat1=1, treat2=1) )

## Calculate transition probabilities from the start of the study to 15 weeks
pmatrix.pieceswise.msm(x, 0, 15, times, covariates)
## End(Not run)
```

---

```
prevalence.msm      Tables of observed and expected prevalences
```

---

**Description**

This provides a rough indication of the goodness of fit of a multi-state model, by estimating the observed numbers of individuals occupying each state at a series of times, and comparing these with forecasts from the fitted model.

**Usage**

```
prevalence.msm(x, times, timezero=NULL, initstates, covariates="mean",
               misccovariates="mean")
```

**Arguments**

|          |                                                                                                                                                   |
|----------|---------------------------------------------------------------------------------------------------------------------------------------------------|
| x        | A fitted multi-state model produced by <a href="#">msm</a> .                                                                                      |
| times    | Series of times at which to compute the observed and expected prevalences of states.                                                              |
| timezero | Initial time of the Markov process. Expected values are forecasted from here. Defaults to the minimum of the observation times given in the data. |

|                             |                                                                                                                                                                                                                 |
|-----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>initstates</code>     | Optional vector of the same length as the number of states. Gives the numbers of individuals occupying each state at the initial time. The default is those observed in the data.                               |
| <code>covariates</code>     | Covariate values for which to forecast expected state occupancy. See <code>qmatrix.msm</code> . Defaults to the mean values of the covariates in the data set.                                                  |
| <code>misccovariates</code> | (Misclassification models only) Values of covariates on the misclassification probability matrix for which to forecast expected state occupancy. Defaults to the mean values of the covariates in the data set. |

## Details

To compute ‘observed’ prevalences at a time  $t$ , individuals are assumed to be in the same state as at their last observation time preceding  $t$ .

The fitted transition probability matrix is used to forecast expected prevalences from the state occupancy at the initial time. To produce the expected number in state  $j$  at time  $t$  after the start, the number of individuals under observation at time  $t$  (including those who have died, but not those lost to follow-up) is multiplied by the product of the proportion of individuals in each state at the initial time and the transition probability matrix in the time interval  $t$ . The proportion of individuals in each state at the "initial" time is estimated, if necessary, in the same way as the observed prevalences.

For misclassification models (fitted using an `ematrix`), this aims to assess the fit of the full model for the *observed* states. That is, the combined Markov progression model for the true states and the misclassification model. Thus, expected prevalences of *true* states are estimated from the assumed proportion occupying each state at the initial time using the fitted transition probability matrix. The vector of expected prevalences of true states is then multiplied by the fitted misclassification probability matrix to obtain the expected prevalences of *observed* states.

For general hidden Markov models, the observed state is taken to be the predicted underlying state from the Viterbi algorithm (`viterbi.msm`). The goodness of fit of these states to the underlying Markov model is tested.

For an example of this approach, see Gentleman *et al.* (1994).

## Value

A list with components:

|                      |                                                                     |
|----------------------|---------------------------------------------------------------------|
| Observed             | Table of observed numbers of individuals in each state at each time |
| Observed percentages | Corresponding percentage of the individuals at risk at each time.   |
| Expected             | Table of corresponding expected numbers.                            |
| Expected percentages | Corresponding percentage of the individuals at risk at each time.   |

## Author(s)

C. H. Jackson <chris.jackson@imperial.ac.uk>

## References

Gentleman, R.C., Lawless, J.F., Lindsey, J.C. and Yan, P. Multi-state Markov models for analysing incomplete disease history data with illustrations for HIV disease. *Statistics in Medicine* (1994) 13(3): 805–821.

## See Also

[msm](#), [summary.msm](#)

---

psor

*Psoriatic arthritis data*

---

## Description

A series of observations of grades of psoriatic arthritis, as indicated by numbers of damaged joints.

## Usage

```
data(psor)
```

## Format

A data frame containing 806 observations, representing visits to a psoriatic arthritis (PsA) clinic from 305 patients. The rows are grouped by patient number and ordered by examination time. Each row represents an examination and contains additional covariates.

|          |             |                                                                                                                             |
|----------|-------------|-----------------------------------------------------------------------------------------------------------------------------|
| ptnum    | (numeric)   | Patient identification number                                                                                               |
| months   | (numeric)   | Examination time in months                                                                                                  |
| state    | (numeric)   | Clinical state of PsA. Patients in states 1, 2, 3 and 4 have 0, 1 to 4, 5 to 9 and 10 or more damaged joints, respectively. |
| hieffusn | (numeric)   | Presence of five or more effusions                                                                                          |
| ollwsdrt | (character) | Erythrocyte sedimentation rate of less than 15 mm/h                                                                         |

## References

Gladman, D. D. and Farewell, V.T. (1999) Progression in psoriatic arthritis: role of time-varying clinical indicators. *J. Rheumatol.* 26(11):2409-13

## Examples

```
## Four-state progression-only model with high effusion and low
## sedimentation rate as covariates on the progression rates. High
## effusion is assumed to have the same effect on the 1-2, 2-3, and 3-4
## progression rates, while low sedimentation rate has the same effect
## on the 1-2 and 2-3 intensities, but a different effect on the 3-4.
```

```
data(psor)
```

```

psor.q <- rbind(c(0,0.1,0,0),c(0,0,0.1,0),c(0,0,0,0.1),c(0,0,0,0))
psor.msm <- msm(state ~ months, subject=ptnum, data=psor,
               qmatrix = psor.q, covariates = ~ollwsdrt+hieffusn,
               constraint = list(hieffusn=c(1,1,1),ollwsdrt=c(1,1,2)),
               fixedpars=FALSE, control = list(REPORT=1,trace=2), method="BFGS")
qmatrix.msm(psor.msm)
sojourn.msm(psor.msm)
hazard.msm(psor.msm)

```

---

qmatrix.msm

*Transition intensity matrix*


---

### Description

Extract the estimated transition intensity matrix, and the corresponding standard errors, from a fitted multi-state model at a given set of covariate values.

### Usage

```
qmatrix.msm(x, covariates="mean", sojourn=FALSE, cl=0.95)
```

### Arguments

|            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| x          | A fitted multi-state model, as returned by <a href="#">msm</a>                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| covariates | The covariate values at which to estimate the intensity matrix. This can either be:<br><br>the string "mean", denoting the means of the covariates in the data (this is the default),<br><br>the number 0, indicating that all the covariates should be set to zero,<br><br>or a list of values, with optional names. For example<br>list (60, 1)<br>where the order of the list follows the order of the covariates originally given in the model formula, or a named list,<br>list (age = 60, sex = 1) |
| sojourn    | Set to TRUE if the estimated sojourn times and their standard errors should also be returned.                                                                                                                                                                                                                                                                                                                                                                                                            |
| cl         | Width of the symmetric confidence interval to present. Defaults to 0.95.                                                                                                                                                                                                                                                                                                                                                                                                                                 |

### Details

Transition intensities and covariate effects are estimated on the log scale by [msm](#). A covariance matrix is estimated from the Hessian of the maximised log-likelihood. The delta method is used to obtain from these the standard error of the intensities on the natural scale at arbitrary covariate values. Confidence limits are calculated by assuming normality on the log scale.

**Value**

A list with components:

|          |                                            |
|----------|--------------------------------------------|
| estimate | Estimated transition intensity matrix.     |
| SE       | Corresponding approximate standard errors. |
| L        | Lower confidence limits                    |
| U        | Upper confidence limits                    |

If `sojourn` is `TRUE`, extra components called `sojourn` and `sojournSE` are included, containing the estimate and standard errors, respectively, of the mean sojourn times in each transient state.

The default print method for objects returned by `qmatrix.msm` presents estimates and confidence limits. To present estimates and standard errors, do something like

```
qmatrix.msm(x) [c("estimates", "SE")]
```

**Author(s)**

C. H. Jackson <chris.jackson@imperial.ac.uk>

**See Also**

`pmatrix.msm`, `sojourn.msm`, `deltamethod`, `ematrix.msm`

---

qratio.msm

*Estimated ratio of transition intensities*

---

**Description**

Compute the estimate and approximate standard error of the ratio of two estimated transition intensities from a fitted multi-state model at a given set of covariate values.

**Usage**

```
qratio.msm(x, ind1, ind2, covariates = "mean", cl = 0.95)
```

**Arguments**

|      |                                                                                                                                 |
|------|---------------------------------------------------------------------------------------------------------------------------------|
| x    | A fitted multi-state model, as returned by <code>msm</code>                                                                     |
| ind1 | Pair of numbers giving the indices in the intensity matrix of the numerator of the ratio, for example, <code>c(1, 2)</code> .   |
| ind2 | Pair of numbers giving the indices in the intensity matrix of the denominator of the ratio, for example, <code>c(2, 1)</code> . |

`covariates` The covariate values at which to estimate the intensities. This can either be:

- the string "mean", denoting the means of the covariates in the data (this is the default),
- the number 0, indicating that all the covariates should be set to zero,
- or a list of values, with optional names. For example
 

```
list (60, 1)
```

 where the order of the list follows the order of the covariates originally given in the model formula, or a named list,
 

```
list (age = 60, sex = 1)
```

`c1` Width of the symmetric confidence interval to present. Defaults to 0.95.

### Details

For example, we might want to compute the ratio of the progression rate and recovery rate for a fitted model `disease.msm` with a health state (state 1) and a disease state (state 2). In this case, the progression rate is the (1,2) entry of the intensity matrix, and the recovery rate is the (2,1) entry. Thus to compute this ratio with covariates set to their means, we call

```
qratio.msm(disease.msm, c(1,2), c(2,1)) .
```

Standard errors are estimated by the delta method. Confidence limits are estimated by assuming normality on the log scale.

### Value

A named vector with elements `estimate`, `se`, `L` and `U` containing the estimate, standard error, lower and upper confidence limits, respectively, of the ratio of intensities.

### Author(s)

C. H. Jackson <chris.jackson@imperial.ac.uk>

### See Also

[qmatrix.msm](#)

---

sim.msm

*Simulate one individual trajectory from a continuous-time Markov model*

---

### Description

Simulate one realisation from a continuous-time Markov process up to a given time.



**Usage**

```
sim.msm(qmatrix, maxtime, covs=NULL, beta=NULL, obstimes=0, start=1,
mintime=0)
```

**Arguments**

|                       |                                                                                                                                                                                                                                                                                                                                                        |
|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>qmatrix</code>  | The transition intensity matrix of the Markov process. The diagonal of <code>qmatrix</code> is ignored, and computed as appropriate so that the rows sum to zero. For example, a possible <code>qmatrix</code> for a three state illness-death model with recovery is:<br><br><code>rbind( c( 0, 0.1, 0.02 ), c( 0.1, 0, 0.01 ), c( 0, 0, 0 ) )</code> |
| <code>maxtime</code>  | Maximum time for the simulated process.                                                                                                                                                                                                                                                                                                                |
| <code>covs</code>     | Matrix of time-dependent covariates, with one row for each observation time and one column for each covariate.                                                                                                                                                                                                                                         |
| <code>beta</code>     | Matrix of linear covariate effects on log transition intensities. The rows correspond to different covariates, and the columns to the transition intensities. The intensities are ordered by reading across rows of the intensity matrix, starting with the first, counting the positive off-diagonal elements of the matrix.                          |
| <code>obstimes</code> | Vector of times at which the covariates are observed.                                                                                                                                                                                                                                                                                                  |
| <code>start</code>    | Starting state of the process. Defaults to 1.                                                                                                                                                                                                                                                                                                          |
| <code>mintime</code>  | Starting time of the process. Defaults to 0.                                                                                                                                                                                                                                                                                                           |

**Details**

The effect of time-dependent covariates on the transition intensity matrix for an individual is determined by assuming that the covariate is a step function which remains constant in between the individual's observation times.

**Value**

A list with components,

|                      |                                                                                                                                                                    |
|----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>states</code>  | Simulated states through which the process moves. This ends with either an absorption before <code>obstime</code> , or a transient state at <code>obstime</code> . |
| <code>times</code>   | Exact times at which the process changes to the corresponding states                                                                                               |
| <code>qmatrix</code> | The given transition intensity matrix                                                                                                                              |

**Author(s)**

C. H. Jackson <chris.jackson@imperial.ac.uk>

**See Also**

[simmulti.msm](#)

## Examples

```
qmatrix <- rbind(
  c(-0.2, 0.1, 0.1 ),
  c(0.5, -0.6, 0.1 ),
  c(0, 0, 0)
)
sim.msm(qmatrix, 30)
```

---

|              |                                                                                                        |
|--------------|--------------------------------------------------------------------------------------------------------|
| simmulti.msm | <i>Simulate multiple trajectories from a multi-state Markov model with arbitrary observation times</i> |
|--------------|--------------------------------------------------------------------------------------------------------|

---

## Description

Simulate a number of individual realisations from a multi-state Markov process. Observations of the process are made at specified arbitrary times for each individual.

## Usage

```
simmulti.msm(data, qmatrix, covariates=NULL, death = FALSE, start,
  ematrix=NULL, hmodel=NULL, hcovariates=NULL)
```

## Arguments

|            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| data       | A data frame with a mandatory column named <code>time</code> , representing observation times. The optional column named <code>subject</code> , corresponds to subject identification numbers. If not given, all observations are assumed to be on the same individual. Observation times should be sorted within individuals. Other named columns of the data frame represent any covariates.                                                                                                                                                    |
| qmatrix    | The transition intensity matrix of the Markov process, with any covariates set to zero. The diagonal of <code>qmatrix</code> is ignored, and computed as appropriate so that the rows sum to zero. For example, a possible <code>qmatrix</code> for a three state illness-death model with recovery is:<br><pre> rbind( c( 0, 0.1, 0.02 ), c( 0.1, 0, 0.01 ), c( 0, 0, 0 ) )</pre>                                                                                                                                                                |
| covariates | List of covariate effects on log transition intensities. Each element is a vector of the effects of one covariate on all the transition intensities. The intensities are ordered by reading across rows of the intensity matrix, starting with the first, counting the positive off-diagonal elements of the matrix.<br>For example, for a multi-state model with three transition intensities, and two covariates <code>x</code> and <code>y</code> on each intensity,<br><pre> covariates=list(x = c(-0.3,-0.3,-0.3), y=c(0.1, 0.1, 0.1))</pre> |

|             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| death       | Vector of indices of the death states. A death state is an absorbing state whose time of entry is known exactly, but the individual is assumed to be in an unknown transient state ("alive") at the previous instant. This is the usual situation for times of death in chronic disease monitoring data. For example, if you specify <code>death = c(4, 5)</code> then states 4 and 5 are assumed to be death states.<br><code>death = TRUE</code> indicates that the final state is a death state, and <code>death = FALSE</code> (the default) indicates that there is no death state. |
| start       | A vector with the same number of elements as there are distinct subjects in the data, giving the states in which each corresponding individual begins. Defaults to state 1 for each subject.                                                                                                                                                                                                                                                                                                                                                                                             |
| ematrix     | An optional misclassification matrix for generating observed states conditionally on the simulated true states. As defined in <a href="#">msm</a> .                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| hmodel      | An optional hidden Markov model for generating observed outcomes conditionally on the simulated true states. As defined in <a href="#">msm</a> .                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| hcovariates | List of the same length as <code>hmodel</code> , defining any covariates governing the hidden Markov outcome models. Unlike in the <code>msm</code> function, this should also define the values of the covariate effects. Each element of the list is a named vector of the initial values for each set of covariates for that state. For example, for a three-state hidden Markov model with two, one and no covariates on the state 1, 2 and 3 outcome models respectively,<br><pre>hcovariates = list (c(acute=-8, age=0), c(acute=-8), NULL)</pre>                                  |

## Details

`sim.msm` is called repeatedly to produce a simulated trajectory for each individual. The state at each specified observation time is then taken to produce a new column `state`. The effect of time-dependent covariates on the transition intensity matrix for an individual is determined by assuming that the covariate is a step function which remains constant in between the individual's observation times. If the subject enters an absorbing state, then only the first observation in that state is kept in the data frame. Rows corresponding to future observations are deleted. The entry times into states given in `death` are assumed to be known exactly.

## Value

A data frame with columns,

|         |                                                                                                         |
|---------|---------------------------------------------------------------------------------------------------------|
| subject | Subject identification indicators                                                                       |
| time    | Observation times                                                                                       |
| state   | Simulated (true) state at the corresponding time                                                        |
| obs     | Observed outcome at the corresponding time, if <code>ematrix</code> or <code>hmodel</code> was supplied |

plus any supplied covariates.

## Author(s)

C. H. Jackson <chris.jackson@imperial.ac.uk>

**See Also**[sim.msm](#)**Examples**

```
### Simulate 100 individuals with common observation times
sim.df <- data.frame(subject = rep(1:100, rep(13,100)), time = rep(seq(0, 24, 2), 100))
qmatrix <- rbind(c(-0.11, 0.1, 0.01 ),
                 c(0.05, -0.15, 0.1 ),
                 c(0.02, 0.07, -0.09))
simmulti.msm(sim.df, qmatrix)
```

sojourn.msm

*Mean sojourn times from a multi-state model***Description**

Estimate the mean sojourn times in the transient states of a multi-state model and their confidence limits.

**Usage**

```
sojourn.msm(x, covariates="mean", cl=0.95)
```

**Arguments**

|                         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|-------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>x</code>          | A fitted multi-state model, as returned by <a href="#">msm</a>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| <code>covariates</code> | The covariate values at which to estimate the mean sojourn times. This can either be: <ul style="list-style-type: none"> <li>the string "mean", denoting the means of the covariates in the data (this is the default),</li> <li>the number 0, indicating that all the covariates should be set to zero,</li> <li>a list of values, with optional names. For example, <pre>list(60, 1)</pre>           where the order of the list follows the order of the covariates originally given in the model formula, or a named list, e.g. <pre>list(age = 60, sex = 1)</pre> </li> </ul> |
| <code>cl</code>         | Width of the symmetric confidence interval to present. Defaults to 0.95.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |

**Details**

The mean sojourn time in a transient state  $r$  is estimated by  $-1/q_{rr}$ , where  $q_{rr}$  is the  $r$ th entry on the diagonal of the estimated transition intensity matrix. Calls [deltamethod](#) to find approximate standard errors. Confidence limits are estimated by assuming normality on the log scale.

**Value**

A data frame with components:

|                        |                                                       |
|------------------------|-------------------------------------------------------|
| <code>estimates</code> | Estimated mean sojourn times in the transient states. |
| <code>SE</code>        | Corresponding standard errors.                        |
| <code>L</code>         | Lower confidence limits.                              |
| <code>U</code>         | Upper confidence limits.                              |

**Author(s)**

C. H. Jackson <chris.jackson@imperial.ac.uk>

**See Also**

[msm](#), [qmatrix.msm](#), [deltamethod](#)

---

`statetable.msm`      *Table of transitions*

---

**Description**

Calculates a frequency table counting the number of times each pair of states were observed in successive observation times. This can be a useful way of summarising multi-state data.

**Usage**

```
statetable.msm(state, subject, data=NULL)
```

**Arguments**

|                      |                                                                                                                                            |
|----------------------|--------------------------------------------------------------------------------------------------------------------------------------------|
| <code>state</code>   | Observed states, assumed to be ordered by time within each subject.                                                                        |
| <code>subject</code> | Subject identification numbers corresponding to <code>state</code> . If not given, all observations are assumed to be on the same subject. |
| <code>data</code>    | An optional data frame in which the variables represented by <code>subject</code> and <code>state</code> can be found.                     |

**Value**

A frequency table with starting states as rows and finishing states as columns.

**Author(s)**

C. H. Jackson <chris.jackson@imperial.ac.uk>

**See Also**

[crudeinits.msm](#)

**Examples**

```
## Heart transplant data
data(heart)

## 148 deaths from state 1, 48 from state 2 and 55 from state 3.
statetable.msm(state, PTNUM, data=heart)
```

---

msm.summary

*Summarise a fitted multi-state model*


---

**Description**

Summary method for fitted [msm](#) models. Currently, this produces a table of observed and expected state prevalences for each time. For models with covariates, prints hazard ratios with confidence intervals for covariate effects.

**Usage**

```
## S3 method for class 'msm':
summary(object, times=NULL, timezero=NULL, initstates=NULL,
covariates="mean", misccovariates="mean", hazard.scale=1, ...)
```

**Arguments**

|                             |                                                                                                                                                                                                                 |
|-----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>object</code>         | A fitted multi-state model object, as returned by <a href="#">msm</a> .                                                                                                                                         |
| <code>times</code>          | A sequence of times at which to compare observed and expected prevalences of each state. Defaults to <code>seq(min(times), max(times), (max(times) - min(times))/10)</code> .                                   |
| <code>timezero</code>       | Initial time of the Markov process. Expected values are forecasted from here. Defaults to the minimum of the observation times given in the data.                                                               |
| <code>initstates</code>     | Optional vector of the same length as the number of states. Gives the numbers of individuals occupying each state at <code>timezero</code> . The default is that all individuals are in state 1.                |
| <code>covariates</code>     | Covariate values for which to forecast expected state occupancy. See <a href="#">qmatrix.msm</a> . Defaults to the mean values of the covariates in the data set.                                               |
| <code>misccovariates</code> | (Misclassification models only) Values of covariates on the misclassification probability matrix for which to forecast expected state occupancy. Defaults to the mean values of the covariates in the data set. |
| <code>hazard.scale</code>   | Vector with same elements as number of covariates on transition rates. Corresponds to the increase in each covariate used to calculate its hazard ratio. Defaults to all 1.                                     |
| <code>...</code>            | further arguments passed to or from other methods                                                                                                                                                               |

**Value**

A list of class `summary.msm`, with components:

`prevalences` Output from `prevalence.msm`.  
`hazard` Output from `hazard.msm`  
`hazard.scale` Value of the `hazard.scale` argument

**Author(s)**

C. H. Jackson <chris.jackson@imperial.ac.uk>

**See Also**

[msm](#), [prevalence.msm](#), [hazard.msm](#)

---

|                          |                                       |
|--------------------------|---------------------------------------|
| <code>surface.msm</code> | <i>Explore the likelihood surface</i> |
|--------------------------|---------------------------------------|

---

**Description**

Plot the log-likelihood surface with respect to two parameters.

**Usage**

```
surface.msm(x, params=c(1,2), np=10, type=c("contour", "filled.contour", "persp", "image"),
            point=NULL, xrange=NULL, yrange=NULL, ...)
```

**Arguments**

|                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |           |                                                           |                  |                                                                              |         |                                                             |         |                                                            |
|---------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|-----------------------------------------------------------|------------------|------------------------------------------------------------------------------|---------|-------------------------------------------------------------|---------|------------------------------------------------------------|
| <code>x</code>      | Output from <code>msm</code> , representing a fitted <code>msm</code> model.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |           |                                                           |                  |                                                                              |         |                                                             |         |                                                            |
| <code>params</code> | Integer vector with two elements, giving the indices of the parameters to vary. All other parameters will be fixed. Defaults to <code>c(1, 2)</code> , representing the first two log transition intensities. See the <code>fixedpars</code> argument to <code>msm</code> for a definition of these indices.                                                                                                                                                                                                                                                                            |           |                                                           |                  |                                                                              |         |                                                             |         |                                                            |
| <code>np</code>     | Number of grid points to use in each direction, by default 10. An <code>np × np</code> grid will be used to evaluate the likelihood surface. If 100 likelihood function evaluations is slow, then reduce this.                                                                                                                                                                                                                                                                                                                                                                          |           |                                                           |                  |                                                                              |         |                                                             |         |                                                            |
| <code>type</code>   | Character string specifying the type of plot to produce. <table style="width: 100%; border-collapse: collapse; margin-top: 5px;"> <tr> <td style="width: 15%;">"contour"</td> <td>Contour plot, using the R function <code>contour</code>.</td> </tr> <tr> <td>"filled.contour"</td> <td>Solid-color contour plot, using the R function <code>filled.contour</code>.</td> </tr> <tr> <td>"persp"</td> <td>Perspective plot, using the R function <code>persp</code>.</td> </tr> <tr> <td>"image"</td> <td>Grid color plot, using the R function <code>image</code>.</td> </tr> </table> | "contour" | Contour plot, using the R function <code>contour</code> . | "filled.contour" | Solid-color contour plot, using the R function <code>filled.contour</code> . | "persp" | Perspective plot, using the R function <code>persp</code> . | "image" | Grid color plot, using the R function <code>image</code> . |
| "contour"           | Contour plot, using the R function <code>contour</code> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |           |                                                           |                  |                                                                              |         |                                                             |         |                                                            |
| "filled.contour"    | Solid-color contour plot, using the R function <code>filled.contour</code> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |           |                                                           |                  |                                                                              |         |                                                             |         |                                                            |
| "persp"             | Perspective plot, using the R function <code>persp</code> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |           |                                                           |                  |                                                                              |         |                                                             |         |                                                            |
| "image"             | Grid color plot, using the R function <code>image</code> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |           |                                                           |                  |                                                                              |         |                                                             |         |                                                            |
| <code>point</code>  | Vector of length <code>n</code> , where <code>n</code> is the number of parameters in the model, including                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |           |                                                           |                  |                                                                              |         |                                                             |         |                                                            |

|                     |                                                                                                                                                                                                                                |
|---------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                     | the parameters that will be varied here. This specifies the point at which to fix the likelihood. By default, this is the maximum likelihood estimates stored in the fitted model <code>x</code> , <code>x\$estimates</code> . |
| <code>xrange</code> | Range to plot for the first varied parameter. Defaults to plus and minus two standard errors, obtained from the Hessian at the maximum likelihood estimate.                                                                    |
| <code>yrange</code> | Range to plot for the second varied parameter. Defaults to plus and minus two standard errors, obtained from the Hessian at the maximum likelihood estimate.                                                                   |
| <code>...</code>    | Further arguments to be passed to the plotting function.                                                                                                                                                                       |

### Details

Draws a contour or perspective plot. Useful for diagnosing irregularities in the likelihood surface. If you want to use these plots before running the maximum likelihood estimation, then just run `msm` with all estimates fixed at their initial values.

`contour.msm` just calls `surface.msm` with `type = "persp"`.

`persp.msm` just calls `surface.msm` with `type = "persp"`.

`image.msm` just calls `surface.msm` with `type = "persp"`.

As these three functions are methods of the generic functions `contour`, `persp` and `image`, they can be invoked as `contour(x)`, `persp(x)` or `image(x)`, where `x` is a fitted `msm` object.

### Author(s)

C. H. Jackson <chris.jackson@imperial.ac.uk>

### See Also

[msm](#), [contour](#), [filled.contour](#), [persp](#), [image](#).

---

tnorm

*Truncated Normal distribution*

---

### Description

Density, distribution function, quantile function and random generation for the truncated Normal distribution with mean equal to `mean` and standard deviation equal to `sd` before truncation, and truncated on the interval `[lower, upper]`.

### Usage

```
dtnorm(x, mean=0, sd=1, lower=-Inf, upper=Inf, log = FALSE)
ptnorm(q, mean=0, sd=1, lower=-Inf, upper=Inf, lower.tail = TRUE, log.p = FALSE)
qtnorm(p, mean=0, sd=1, lower=-Inf, upper=Inf, lower.tail = TRUE, log.p = FALSE)
rtnorm(n, mean=0, sd=1, lower=-Inf, upper=Inf)
```



**Arguments**

|                         |                                                                                                           |
|-------------------------|-----------------------------------------------------------------------------------------------------------|
| <code>x, q</code>       | vector of quantiles.                                                                                      |
| <code>p</code>          | vector of probabilities.                                                                                  |
| <code>n</code>          | number of observations. If <code>length(n) &gt; 1</code> , the length is taken to be the number required. |
| <code>mean</code>       | vector of means.                                                                                          |
| <code>sd</code>         | vector of standard deviations.                                                                            |
| <code>lower</code>      | lower truncation point.                                                                                   |
| <code>upper</code>      | upper truncation point.                                                                                   |
| <code>log, log.p</code> | logical; if TRUE, probabilities <code>p</code> are given as <code>log(p)</code> .                         |
| <code>lower.tail</code> | logical; if TRUE (default), probabilities are $P[X \leq x]$ , otherwise, $P[X > x]$ .                     |

**Details**

The truncated normal distribution has density

$$f(x, \mu, \sigma) = \phi(x, \mu, \sigma) / (\Phi(u, \mu, \sigma) - \Phi(l, \mu, \sigma))$$

for  $l \leq x \leq u$ , and 0 otherwise.

$\mu$  is the mean of the original Normal distribution before truncation,

$\sigma$  is the corresponding standard deviation,

$u$  is the upper truncation point,

$l$  is the lower truncation point,

$\phi(x)$  is the density of the corresponding normal distribution, and

$\Phi(x)$  is the distribution function of the corresponding normal distribution.

If `mean` or `sd` are not specified they assume the default values of 0 and 1, respectively.

If `lower` or `upper` are not specified they assume the default values of `-Inf` and `Inf`, respectively, corresponding to no lower or no upper truncation.

Therefore, for example, `dtnorm(x)`, with no other arguments, is simply equivalent to `dnorm(x)`.

Only `rtnorm` is used in the `msm` package, to simulate from hidden Markov models with truncated normal distributions. These functions are merely provided for completion, and are not optimized for numerical stability. To fit a hidden Markov model with a truncated Normal response distribution, use a `hmmTNorm` constructor. See the `hmm-dists` help page for further details.

**Value**

`dtnorm` gives the density, `ptnorm` gives the distribution function, `qtnorm` gives the quantile function, and `rtnorm` generates random deviates.

**Author(s)**

C. H. Jackson <chris.jackson@imperial.ac.uk>

**See Also**

[dnorm](#)

**Examples**

```
x <- seq(50, 90, by=1)
plot(x, dnorm(x, 70, 10), type="l", ylim=c(0,0.06)) ## standard Normal distribution
lines(x, dtnorm(x, 70, 10, 60, 80), type="l")      ## truncated Normal distribution
```

---

|            |                             |
|------------|-----------------------------|
| totlos.msm | <i>Total length of stay</i> |
|------------|-----------------------------|

---

**Description**

Estimate the expected total length of stay in each transient state, for a given period of evolution of a multi-state model. This assumes that the transition rates do not change with time.

**Usage**

```
totlos.msm(x, start=1, fromt=0, tot=Inf, covariates="mean", ci.boot=FALSE, cl=0.95,
```

**Arguments**

|            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| x          | A fitted multi-state model, as returned by <a href="#">msm</a> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| start      | State at the beginning of the period.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| fromt      | Time from which to estimate total length of stay. Defaults to 0, the beginning of the process.                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| tot        | Time up to which total length of stay is estimated. Defaults to infinity, giving the expected time spent in the state until absorption. For models without an absorbing state, $\tau$ must be specified.                                                                                                                                                                                                                                                                                                                                              |
| covariates | The covariate values to estimate for. This can either be: <ul style="list-style-type: none"> <li>the string "mean", denoting the means of the covariates in the data (this is the default),</li> <li>the number 0, indicating that all the covariates should be set to zero,</li> <li>or a list of values, with optional names. For example <pre>list (60, 1)</pre>           where the order of the list follows the order of the covariates originally given in the model formula, or a named list, <pre>list (age = 60, sex = 1)</pre> </li> </ul> |
| ci.boot    | Calculate a bootstrap confidence interval. This is usually time-consuming, and disabled by default. See <a href="#">boot.msm</a> for more details of bootstrapping in <b>msm</b> .                                                                                                                                                                                                                                                                                                                                                                    |
| cl         | Width of the symmetric confidence interval                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| B          | Number of bootstrap replicates                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| ...        | Further arguments to be passed to the <a href="#">integrate</a> function to control the numerical integration.                                                                                                                                                                                                                                                                                                                                                                                                                                        |

**Details**

The expected total length of stay in state  $j$  between times  $t_1$  and  $t_2$ , from the point of view of an individual in state  $i$  at time 0, is defined by the integral from  $t_1$  to  $t_2$  of the  $i, j$  entry of the transition probability matrix  $P(t)$ . As the individual entries of  $P(t) = \exp(tQ)$  are not available explicitly in terms of  $t$  for a general Markov model, this integral is calculated numerically, using the `integrate` function. This may take a long time for models with many states where  $P(t)$  is expensive to calculate.

For a model where the individual has only one place to go from each state, and each state is visited only once, for example a progressive disease model with no recovery or death, these are equal to the mean sojourn time in each state. However, consider a three-state health-disease-death model with transitions from health to disease, health to death, and disease to death, where everybody starts healthy. In this case the mean sojourn time in the disease state will be greater than the expected length of stay in the disease state. This is because the mean sojourn time in a state is conditional on entering the state, whereas the expected total time diseased is a forecast for a healthy individual, who may die before getting the disease.

**Value**

A vector of expected total lengths of stay for each transient state.

**Author(s)**

C. H. Jackson <chris.jackson@imperial.ac.uk>

**See Also**

`sojourn.msm`, `pmatix.msm`, `integrate`, `boot.msm`.

---

`transient.msm`      *Transient and absorbing states*

---

**Description**

Returns the transient and absorbing states of either a fitted model or a transition intensity matrix.

**Usage**

```
transient.msm(x=NULL, qmatrix=NULL)
absorbing.msm(x=NULL, qmatrix=NULL)
```

**Arguments**

`x`                    A fitted multi-state model as returned by `msm`.

`qmatrix`            A transition intensity matrix. The diagonal is ignored and taken to be minus the sum of the rest of the row.

**Value**

A vector of the ordinal indices of the transient or absorbing states.

**Author(s)**

C. H. Jackson <chris.jackson@imperial.ac.uk>

---

viterbi.msm

*Calculate the most likely path through underlying stages*

---

**Description**

For a fitted hidden Markov model, the Viterbi algorithm recursively constructs the path with the highest probability through the underlying stages.

**Usage**

```
viterbi.msm(x)
```

**Arguments**

`x` A fitted hidden Markov multi-state model, as produced by [msm](#)

**Value**

A data frame with columns:

`subject` = subject identification numbers

`time` = times of observations

`observed` = corresponding observed states

`fitted` = corresponding fitted states found by Viterbi recursion. If the model is not a hidden Markov model, this is just the observed states.

**Author(s)**

C. H. Jackson <chris.jackson@imperial.ac.uk>

**References**

Durbin, R., Eddy, S., Krogh, A. and Mitchison, G. *Biological sequence analysis*, Cambridge University Press, 1998.

**See Also**

[msm](#)

# Index

## \*Topic **datasets**

aneur, 3  
bos, 5  
fev, 11  
heart, 13  
psor, 36

## \*Topic **math**

deltamethod, 8  
MatrixExp, 1

## \*Topic **models**

coef.msm, 6  
ematrix.msm, 10  
hazard.msm, 12  
logLik.msm, 16  
msm, 19  
odds.msm, 29  
plot.msm, 31  
prevalence.msm, 35  
qmatrix.msm, 37  
qratio.msm, 39  
sim.msm, 40  
simmulti.msm, 41  
sojourn.msm, 43  
surface.msm, 47  
viterbi.msm, 52

absorbing.msm (*transient.msm*), 51  
aneur, 3

boot.msm, 4, 4, 32, 33, 50, 51  
bos, 5

coef.msm, 6  
contour, 47, 48  
contour.msm (*surface.msm*), 47  
crudeinits.msm, 7, 45

dbinom, 15  
deltamethod, 8, 39, 44  
dexp, 15, 30

dgamma, 15  
dmenorm (*medists*), 17  
dmeunif (*medists*), 17  
dnbinom, 15  
dnorm, 19, 49  
dpexp (*pexp*), 29  
dpois, 15  
dtnorm, 19  
dtnorm (*tnorm*), 48  
dunif, 19  
dweibull, 15

ematrix.msm, 10, 11, 26, 39

fev, 11  
filled.contour, 47, 48

hazard.msm, 12, 29, 46, 47  
heart, 13  
hmm-dists, 18, 21, 49  
hmm-dists, 14  
hmmBinom (*hmm-dists*), 14  
hmmCat, 21  
hmmCat (*hmm-dists*), 14  
hmmExp (*hmm-dists*), 14  
hmmGamma (*hmm-dists*), 14  
hmmIdent (*hmm-dists*), 14  
hmmLNorm (*hmm-dists*), 14  
hmmMETNorm, 18  
hmmMETNorm (*hmm-dists*), 14  
hmmMEUnif, 18  
hmmMEUnif (*hmm-dists*), 14  
hmmNBinom (*hmm-dists*), 14  
hmmNorm (*hmm-dists*), 14  
hmmPois (*hmm-dists*), 14  
hmmTNorm, 49  
hmmTNorm (*hmm-dists*), 14  
hmmUnif (*hmm-dists*), 14  
hmmWeibull (*hmm-dists*), 14  
image, 47, 48

image.msm (*surface.msm*), 47  
 integrate, 50, 51  
  
 load, 4  
 logLik.msm, 16  
  
 MatrixExp, 1  
 medists, 14, 17  
 msm, 4, 6–8, 10–13, 16, 17, 19, 27, 29, 31–33, 35, 36, 38, 39, 42, 44, 46–48, 50–52  
 msm.summary, 46  
  
 nlm, 25, 27  
  
 odds.msm, 13, 29  
 optim, 24, 25, 27  
  
 persp, 47, 48  
 persp.msm (*surface.msm*), 47  
 pexp, 29  
 plot.msm, 28, 31  
 pmatrix.msm, 4, 5, 27, 28, 32, 33, 34, 39, 51  
 pmatrix.piecewise.msm, 33, 33  
 pmenorm (*medists*), 17  
 pmeunif (*medists*), 17  
 ppexp (*pexp*), 29  
 prevalence.msm, 35, 46, 47  
 print.summary.msm (*msm.summary*), 46  
 psor, 36  
 ptnorm (*tnorm*), 48  
  
 qmatrix.msm, 11, 26–28, 33, 35, 37, 38, 40, 44, 46  
 qmenorm (*medists*), 17  
 qmeunif (*medists*), 17  
 qpexp (*pexp*), 29  
 qratio.msm, 39  
 qtnorm (*tnorm*), 48  
  
 rmenorm (*medists*), 17  
 rmeunif (*medists*), 17  
 rpexp (*pexp*), 29  
 rtnorm (*tnorm*), 48  
  
 save, 4  
 sim.msm, 30, 40, 42, 43  
 simmulti.msm, 28, 41, 41  
 sojourn.msm, 27, 28, 39, 43, 51  
  
 statetable.msm, 8, 45  
 summary.msm, 12, 13, 28, 36  
 summary.msm (*msm.summary*), 46  
 surface.msm, 47  
  
 tnorm, 14, 48  
 totlos.msm, 5, 50  
 transient.msm, 51  
  
 viterbi.msm, 36, 52